Automatic Annotation and Assessment of Syntactic Structures in Law Texts

Combining Rule-Based and Statistical Methods

Thesis
presented to the Faculty of Arts and Social Sciences
of the University of Zurich
for the degree of Doctor of Philosophy

by

Kyoko Sugisaki

Accepted in the spring semester 2016
on the recommendation of the doctoral committee:

Prof. Dr. Martin Volk (main advisor)

PD Dr. Gerold Schneider

Prof. Dr. Sandra Kübler

Zurich, 2016

"Do not hurry, do not rest."

Goethe

Abstract

In this thesis, I investigate and develop methods for automatically analyzing and assessing German syntactic structures in domain-specific texts. As domain-specific texts, I use Swiss German-language law texts.

The automatic annotation of syntactic structures has long been studied in the research on natural language processing. Supervised statistical methods are regarded as state-of-the-art parsing methods, which are accurate but biased by the type of text. Consequently, the accuracy of statistical parsers decreases if they are used on domain-specific texts. The problem of domain bias in syntactic annotation should be solved if it directly affects the accuracy of an application. The syntactic assessment that I develop in this thesis is such an application that requires high accuracy of syntactic annotation. An effective solution to this problem would be the manual annotation of a large portion of the required domain texts. However, it is not feasible in practice because manual linguistic annotation is extremely labor intensive. To overcome this problem, I develop syntactic annotation methods that do not require the manual annotation of a large portion of the domain texts. The goal of this thesis is that the annotation accuracy on domain-specific texts is so high that it can be used for the application.

For the automatic syntactic assessment, I demonstrate a novel approach to model domain-specific style choice by combining rule-based and statistical methods. In the rule-based approach, I present a method that automatically detects the violations of style rules in legislative style guidelines. In the statistical approach, domain-specific writing style is defined in terms of stylistic choice between syntactic alternations. The syntactic selection is statistically modeled by classifying syntactic alternatives according to their syntactic complexity. The syntactic assessment requires automatic syntactic annotation.

For the automatic syntactic annotation, I present a linguistically motivated hybrid supertagger that analyzes topological dependency grammar relations in the German language. In this thesis, supertagging problems are seen as morphosyntactic ambiguity and syntactic resolution. Depending on the linguistic phenomena, the ambiguity is resolved by applying a rule-based and statistical tagging method: Morphological and syntactic hard constraints are applied in a constraint grammar approach. In contrast, lexical, semantic, and pragmatic soft and multivariate constraints are integrated into a conditional random fields model.

The main contribution of this thesis to the study of natural language processing is to show that a linguistically motivated annotation method is a viable approach to achieving a high performance of syntactic analysis with a few hundreds of manually annotated sentences from the domain.

Acknowledgements

First and foremost, I would like to thank my supervisors and defense committee Martin Volk, Gerold Schneider and Sandra Kübler for their insightful advice. I would like to thank Michael Hess for offering me the fantastic opportunity to pursue a PhD at the university of Zurich. I wish to express my deepest thanks to Stefan Höfler for his constant guidance through my PhD. Without his supports, I would not have had the courage to pursue nor to continue my PhD in computational linguistics as a linguist. Many thanks also to Simon Clematide for his technical suggestions and hints.

I wish to thank all members of the Institute of Computational Linguistics and the technical support team of the Institute of Informatics at the University of Zurich. A special thank goes to Rico Sennrich and Michi Amsler for all the technical discussions and support during the first year of my PhD study. Special thank also goes to Alexandra Bünzli for many administrative and personal support and Annette Rios for reading the drafts of the thesis. Last but not least, I'd like to thank all CL members for daily talks and discussions at lunch times and coffee breaks. Our daily communication was the source of ideas, motivations and energy for research.

I am grateful to the Swiss National Science Foundation that funded my first three years of my PhD in the project "Automated Detection of Style Guide Violations in Legislative Drafts" (grant nr. 134701) and to the committee of the Forschungskredit Candoc at the University of Zurich that funded my fourth year. I wish to thank Rebekka Bratschi in the Swiss Federal Chancellery for the wonderful collaboration.

Finally, but not at least, I would like to express my gratitude to my family and friends, in particular, to my husband for being so supportive and standing by me in many small and big crises during my PhD.

To my family

Contents

List of Figures

List of Tables

Abbreviations

Technical terms

CG	Constraint grammer
CRF	Conditional random field(s)
HMM	Hidden Markov model
MaxEnt	Maximal entropy
MEMM	Maximum entropy Markov model
ML	Machine learning
NLP	Natural language processing
SVM	Support vector machine
XML	Extensible markup language

General linguistic terms

COMP	Complementizer
CONJ	Subordinating conjunction
CP	Complementizer phrase
FIN	Finite verb
ICONJ	Infinitive conjunction
LB	Left bracket
NONFIN	Nonfinite verb
NP	Noun phrase
POS	Part of speech
PP	Prepositional phrase
PUNCT	Punctuation
RB	Right bracket
RC-PRON	Relative pronoun
ZU-INF	Zu-infinitive

Topological dependency grammar

ADV	Adverb
ATTR	Attributive
APP	Apposition
DEP=LB-VF-RC	Left bracket of a verb-final clause (relative clause)
DET	Determiner
EXPL	Expletive *es*
GMOD	Genitive modification
GRAD	Grade expression
KOM	Comparative conjunction
LB-V1	Left bracket of a verb-first clause
LB-V2	Left bracket of a verb-second clause
LB-VF-CONJ	Left bracket of a verb-final clause (subordinated clause)
NP2	Subject in an ellipsis coordination structure
LXCOMP	Left bracket of an infinitive clause
OBJA	Accusative (direct) object
OBJA2	Second accusative (direct) object
OBJD	Dative (indirect) object
OBJG	Genitive object
OBJP	Prepositional object
PN	Dependent of a preposition
PRED	Predicate
PP	Preposition (adjunct)
RB-V1	Right bracket of a verb-second clause
RB-V2	Right bracket of a verb-second clause
RB-VF-CONJ	Right bracket of a verb-final clause (subordinated clause)
RB-VF-RC	Right bracket of a verb-final clause (relative clause)
RXCOMP	Right bracket of an infinitive clause
SUBJ	Subject
TOP	Topicalized right-most verb in a verb-second clause
VC	Part of a verbal complex
ZEIT	Time expression

Dependency grammar

ROOT	Root
NEB	Subordinated clause
REL	Relative clause
SUBC	Subject clause
OBJC	Object clause
OBJI	Object nonfinite clause
AUX	Verb complex

Morphological feature (Gertwol)	cf. Appendix A

Part-of-Speech (STTS)	cf. Appendix B

Chapter 1

Introduction

> *Quellenschutz: Verweigern Personen, die sich beruflich mit der Veröffentlichung von Informationen im redaktionellen Teil eines periodisch erscheinenden Mediums befassen, oder ihre Hilfspersonen das Zeugnis über die Identität des Autors oder über Inhalt und Quellen ihrer Informationen, so dürfen weder Strafen noch prozessuale Zwangsmassnahmen gegen sie verhängt werden.*
>
> – Swiss Criminal Code Art. 28a para. 1

1.1 Motivation

Law texts are often criticized as being incomprehensible to non-lawyers (e.g. Müller, 2006, pp. 185; Lötscher, 1995, pp. 109; Hauk and Lötscher, 1994, pp. 92). A layperson might need to read a paragraph such as the previous one several times before its meaning becomes clear. Legal texts are normative, and they describe legal conditions and consequences. Legal conditions are complex because they attempt to include all imaginable cases, while being both general and sufficiently precise. This requirement leads to the excessive use of complex syntactic structures, such as coordination structures, and clausal modifiers, such as relative clauses and subordinate clauses. Legislative language is characterized by long sentences, nominalization, complex morphological derivations, personalization, and archaic words and phrases (Hansen-Schirra and Neumann, 2004, pp. 170).[1] Long sentences and nominalization have been regarded as contributing to the

[1] In Italian and English law texts, unlike newspapers, prepositional phrases are used extensively, whereas verbal phrases are used less often (Venturi, 2008).

complexity of texts (e.g. Hansen-Schirra and Neumann, 2004, Coleman and Blumenfeld, 1963, Coleman, 1962).

To improve the comprehensibility of Swiss law texts, legislative drafts are edited by linguistic and legal experts who improve the quality of the language (cf. Nussbaumer, 2002). In addition to ensuring the consistency of legal terms and references, the editors examine the logical structures of both the texts and the sentences within them. Hence, the editing of law texts is a time-consuming task. Moreover, the number of law texts that require editing has grown significantly in recent decades.[2] Our complex society needs legal rules and regulations more than ever. Therefore, the capability to proofread legislative drafts automatically would both support editors in fulfilling their tasks and save governmental resources. In this thesis, I focus on syntax to investigate a method for automatically identifying passages in texts that violate the desired writing style in the domain. The objective is to develop an effective model for automatically detecting violations of syntax-related rules in legislative style guidelines.

To detect the violations of style in legislative drafts automatically, the texts are augmented with the appropriate linguistic information. In recent years, automatic morphosyntactic analysis (or part-of-speech tagging) has reached market-ready performance because of the development of sophisticated statistical methods and the availability of linguistically annotated data. This method is now considered good enough to be used in several applications and even linguistic corpora (e.g. Deutsches Referenzkorpus DeReCo[3] or COSMAS I[4]). However, there is still room for improvement in the automatic analysis of syntactic functions and structures (i.e. parsing). According to NLP researchers (e.g. Martschat and Strube, 2014, Quirk and Corston-Oliver, 2006), the errors in an application often are not rooted in their own tasks but in the preprocessing parsers used in the process, which is discussed in detail in Chapter 2. One reason that the performance of state-of-the-art statistical parsers is insufficient, showing low accuracy on real data in real applications, is that they are biased by the type of text (Gildea, 2001).[5] The insufficient accuracy of syntactic annotation cannot be ignored in applications that depend

[2] According to the Swiss parliament, the number of new Swiss law texts that went through the parliament doubled in last 20 years according: http://www.parlament.ch/d/dokumentation/statistiken/Seiten/erlasse.aspx (last visited: 2015/09/18). In 2014, this "Gesetzesflut" (a flood of laws) was covered by media such as SwissInfo (http://www.swissinfo.ch/ger/gesetzesflut_die-schweiz-ist-meister-im-erlassen-neuer-gesetze/40542848 (last visited: 2015/09/18)) and Handelszeitung (http://www.handelszeitung.ch/politik/gesetzesflut-wir-regulieren-uns-zu-tode-675925 (last visited: 2015/09/18))

[3] http://www1.ids-mannheim.de/kl/projekte/korpora/ (last visited: 2014/03/31)

[4] http://www1.ids-mannheim.de/kl/projekte/cosmas_i.html(last visited: 2014/03/31)

[5] Statistical parsers perform at best if they are trained and tested on the same corpora. Consequently, they do not perform as expected if they are tested on data from a different domain. In the case of pipeline-based parsers, the part-of-speech taggers used in the process also affect the performance (e.g. Kübler

heavily on the quality of the parsed data, such as a style checking tool. If the preprocessing syntactic annotation is inaccurate, the style checking tool often recognizes the parser's mistakes as style errors (i.e. false positives). In this thesis, I investigate a method of automatic syntactic annotation that is optimized for an application such as a style checking tool.

The syntactic annotation and assessment method that I investigate in this thesis is part of the style checking tool AutoLegis, which was developed during a research project. This style checking tool and the research project are described in the following section.

1.2 Background: Style Checking Tool

The style checking tool AutoLegis is a system that automatically detects the violations of style rules in guidelines. It highlights violations and provides references to the corresponding guideline. AutoLegis was developed in the research project "Automated Detection of Style Guide Violations in Legislative Drafts." This research project was conducted by the Institute of Computational Linguistics at the University of Zurich in collaboration with the Swiss federal administration, where legislative drafts are reviewed and edited.[6] My PhD project was a part of the project. The tool is designed to assist legislative editors in checking and finding the recommendations and restrictions in the style guidelines for the drafting of law texts. A typical output of this style checking tool is illustrated in Figure 1.1. The highlighted sentence contains potential style problems, and the pop-up window provides their specifications.[7]

and Maier, 2013, Curran et al., 2006). Part-of-speech taggers, again, show corpus affect (e.g. Ferraro et al., 2013, Tsuruoka et al., 2005).

[6]The project was initialized by Stefan Höfler. His research ideas formed the aim of the project and the goal of the development of the style checking tool for law texts. The project was supervised by Prof. Dr. Michael Hess at the Institute of Computational Linguistics and Prof. Dr. Felix Uhlmann at the Faculty of law at the University of Zurich and Dr. Rebekka Bratschi at the Swiss federal administration. The project was funded by the Swiss National Science Foundation (05/2011 - 02/2015) and the Forschungskredit *candoc* of the University of Zurich (08/2014 - 07/2015). The information is available at: `http://www.cl.uzh.ch/research/maschinellestilpruefung/gesetzestextanalyse_en.html` (last visited: 2015/09/19).

[7]The description of the style checking tool is based on Höfler and Sugisaki (2012).

Art. 163 Form der Erlasse der Bundesversammlung

1 Die Bundesversammlung erlässt rechtsetzende Bestimmungen in der Form des Bundesgesetzes oder der Verordnung.

2 Die übrigen Erlasse ergehen in der For
beschluss, der dem Referendum nicht unter
bezeichnet.

83 Angenommen in der Volksabstimmung vom
BRB vom 4. Febr. 2002 – AS **2002** 241; BB
84 Angenommen in der Volksabstimmung vom
BRB vom 4. Febr. 2002 – AS **2002** 241; BB

50

Ein Satz – eine Regelungseinheit

Ein Satz - eine Regelungseinheit

Regel	Ein Satz sollte nicht mehr als einen Gedankengang (eine Norm) enthalten.
Erklärung	Das gleichzeitige Vorkommen eines Objekts und einer adverbialen Bestimmung ist oft ein Indikator dafür, dass ein Satz mehr als eine Regelungseinheit enthält.
Beispiele	*Negativbeispiel:* «Die Weiter- und Fortbildung ist Aufgabe der Spitäler unter Aufsicht der Gesundheitsdirektion.» *Besser:* «Die Weiter- und Fortbildung ist Aufgabe der Spitäler. Sie stehen dabei unter Aufsicht der Gesundheitsdirektion.»
Referenz	Weitere Informationen finden Sie hier: GTR Rz. 887, ZH Rz. 252

FIGURE 1.1: User interface for AutoLegis

The software architecture of AutoLegis is shown in Figure 1.2. Legislative drafts are converted into XML. The text then undergoes several NLP pre-processing phases to be augmented with the appropriate linguistic information before the detection rules are applied to the input text. Finally, the system yields error messages with style rules that correspond to each highlighted passage.

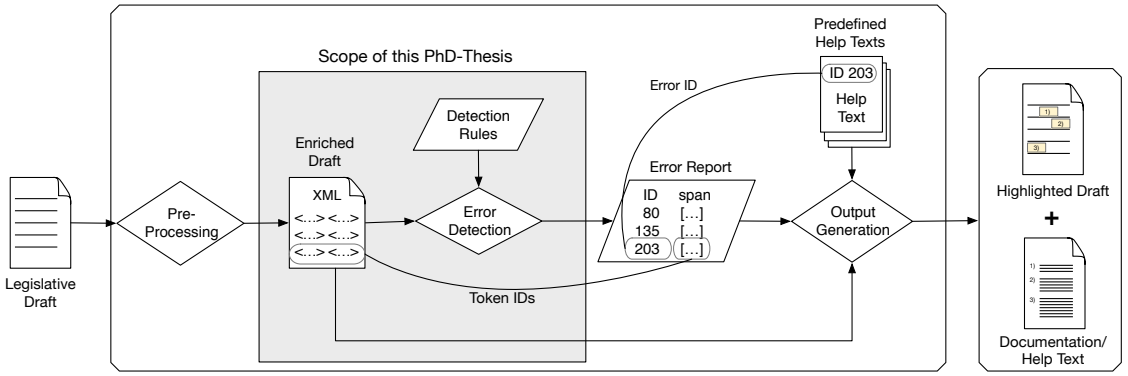


FIGURE 1.2: Software architecture for AutoLegis

AutoLegis detects a wide range of style rules, such as word formation, lexical preference, sentence structure, and document format. In this thesis, I focus on two components of the syntactic analysis performed by AutoLegis: 1) the automatic analysis of syntactic structures and 2) the automatic detection of violations of syntax-related style rules. These two components are briefly described below.

1. **Analysis component: Automatic Syntactic Annotation**
   The first component corresponds to the enrichment of texts in XML by linguistic information in the architecture 1.2. To achieve this aim, texts are automatically

processed by a part-of-speech tagger and a morphological analyzer. Based on the results of the analysis, the texts are then syntactically annotated.

2. **Detection component: Automatic Style Assessment**
   The violations of style guideline rules are automatically detected on the basis of the syntactic information annotated in the first component. For this purpose, I predefine a set of rules to model style error detection in the domain. The component corresponds to detection rules and error detection in the architecture 1.2.

1.3 Research Questions

A main contribution of this thesis is the automatic annotation and assessment of syntactic structures of law texts in German. The main goal of the thesis is the development of methods that can be used for the syntactic annotation and assessment of domain-specific texts with high accuracy even under the constraint of sparse in-domain resources.

Modern statistical (or supervised machine learning) methods need a large amount of linguistically annotated training data. The required manual annotation of training data is labor intensive. Because the type of text of the annotated data affects the performance of the system, the lack of domain-specific training data is a big practical challenge. Therefore, there is an urgent need to design methods that are adopted to the type of text used to build an NLP system in practice. Ideally, domain adaptation methods do not require a large amount of manually annotated in-domain data. Therefore, the following research questions are posed:

- How can a high degree of accuracy be achieved in automatic syntactic annotation in domain-specific texts, such as law texts, with sparse in-domain resources for an application?
- How can we model domain-specific style error detection without annotated data?

I will return to the questions again in the concluding chapter, to discuss the results.

1.4 Thesis Outline

The first and last chapters (i.e., Chapters 2 and 6) of the thesis focus on the automatic detection of violations of syntax-related style rules. In Chapter 2, I develop a style error detection method in a state-of-the-art rule-based approach. In Chapter 6, I propose the

improvements to the style error detection component, using statistical methods. The middle chapters (i.e. Chapters 3, 4, and 5) focus on the improvements of the automatic syntactic analysis.

The chapters of the thesis are organized as follows:

Chapter 2 introduces a rule-based method to automatically detect violations of style rules in guidelines. In this chapter, I use state-of-the-art statistical parsers for automatic syntactic annotation and I discuss how they affect the accuracy of the style error detection. Furthermore, I evaluate the parsers using newspaper and law texts to gain insights into the domain differences in statistical parsing.

Chapter 3 focuses on the improvement of automatic syntactic annotation for the detection of style errors (see Chapter 2). I begin by discussing domain adaptation problems and the peculiarities of the German language—morphosyntactic ambiguity and free word order. I then propose a rule-based and statistical supertagging approach to address these two problems. These methods are then described in detail in Chapters 4 and 5, respectively.

Chapter 4 examines morphosyntactic and syntactic ambiguities in German and presents a rule-based supertagging method to resolve them. I investigate how far this rule-based supertagger can disambiguate morphosyntactic ambiguity and assign syntactic relations.

Chapter 5 focuses on issues of free word order in German and explores the statistical tagging methods used to address them. In this chapter, I explore ways to achieve high accuracy in automatic syntactic annotation with sparse in-domain resources.

Chapter 6 closes the loop in automatic style error detection (see Chapter 2) and evaluates the method using the developed supertagger. In addition, I present a further improvement in style error detection by developing a statistical method to select instances exceeding the syntactic complexity that is conventional in the domain.

Chapter 7 summarizes the results of the thesis and offers outlook for future research.

1.5 Overview of Used Data Sets

In this thesis, I use two types of data sets: Swiss German-language law texts and a German newspaper corpus, TüBa-D/Z.

Below, I present an overview of the data sets:

- **Swiss German-language law texts** are the German version of the Swiss Legislation Corpus (SLC) that was built in the research project "Automated Detection of Style Guide Violations in Legislative Drafts."[8] The SLC is the collection of the contemporary law of Switzerland and consists of the three Swiss national languages (German, French, and Italian). Each language version comprises 1,915 texts. The entire collection of these law texts is available online as the Classified Compilation of Swiss Federal Legislation.[9] To build the SLC, law texts were automatically extracted from the compilation and structured into XML format.[10] The documents are enriched with structures such as title, article, and paragraph (cf. Höfler and Piotrowski, 2011). The German version of the SLC (henceforth Swiss German-language law texts) comprises 4,742,878 tokens (122,857 token types) and 165,188 sentences. The Swiss German-language law texts are the main data set used in this thesis.

- **Tübingen Treebank of Written German (TüBa-D/Z)** is the manually annotated German newspaper corpus based on "Die Tageszeitung".[11] In this thesis, I use version 7 of TüBa-D/Z annotated with dependency grammar relations in CoNLL format. The corpus comprises 65,522 sentences and 1,164,726 tokens (115,351 token types). In Chapter 2, I make use of the TüBa-D/Z data for the training of statistical parsers. In Chapter 5, the TüBa-D/Z data is used for the development of the statistical supertagger.

[8] The description of the SLC is based on Höfler and Piotrowski (2011) and Höfler and Sugisaki (2014)

[9] The Classified Compilation of Swiss Federal Legislation is available in HTML and PDF formats at: https://www.admin.ch/gov/de/start/bundesrecht/systematische-sammlung.html (last visited: 30/07/2015).

[10] This part of the project is conducted by Stefan Höfler.

[11] The information is available at: http://www.sfs.uni-tuebingen.de/en/ascl/resources/corpora/tueba-dz.html (last visited: 2015/09/19)

Chapter 2

Automatic Style Error Detection

Staatsangehörige nach den Absätzen 1 und 4, die entweder Inhaberinnen oder Inhaber einer nicht biometrischen, nach dem 12. Dezember 2008 gemäss den Anforderungen der Verordnung (EG) Nr. 1030/20024 ausgestellten Karte oder eines anderen Dokuments in Papierform sind, können diese Karte oder dieses Dokument bis zum Ablauf der Gültigkeit behalten.

– Regulation on permit, residence and employment Art. 71d para. 6

2.1 Introduction

Law texts should meet three quality requirements: precision, brevity, and simplicity (Müller, 2006, pp. 192–195). In order to meet these requirements, law texts go through several editorial refinements. Text editing is a time-consuming task, and automatic style checkers (or controlled language tools) have been developed since the 1980s (e.g. UNIX Writer's Workbench (MacDonald, 1983), IBM's CRITIQUE (Ravin, 1988), MultiLint (Schmidt-Wigger, 1998), FLAG (Bredenkamp et al., 2000) and Check-Point (Crysmann et al., 2008)). In these systems, style errors are modeled by anticipating error types. For this purpose, undesirable style rules are pre-defined based on expert knowledge. The style rules are then searched automatically. The UNIX Writer's Workbench is a typical style checking tool that is comprised of handcrafted rules based on style guidelines, writing standards in rhetorical traditions, and the results of psychological and linguistic research (Frase, 1983). In addition, style checking tools are often limited to a certain sub-language, such as technical documentation (e.g. MultiLint). A domain-specific style-checking tool has the advantage that the style rules can be tailored to the type of text used in a domain-specific language. In MultiLint, style rules are applied to either

informative text types or instructive text types in technical documents. For example, the rule "always write complete sentences, and do not suppress verbs" is only applied to informative texts; consequently, a sentence such as "*Wärmetauscher undicht?*" is marked in informative texts but not in instructive texts (Schmidt-Wigger, 1998).

In this chapter, I implement a rule-based domain-specific error detection system that automatically detects syntax-related violations of style guideline in law texts. As discussed in Chapter 1, the detection system is embodied in the style checking tool. The detected errors are presented as potential problems and are not automatically corrected. Therefore, the goal is to develop an error detection method with as few false positives as possible. Methodologically, style errors are modeled by anticipating specific linguistic properties that represent style violations. In the implementation, the pre-defined style violations are identified based on a rule-based method. This rule-based error-modeling is a state-of-the-art method and does not require a large number of manual annotations for its development. In this thesis, I use domain-specific style guidelines to define problematic types of syntactic structures, thereby adapting the state-of-the-art error-modeling method to the legislative domain. Because the style rules in these guidelines are often not specific enough to be modeled for the automated error detection, I further concretize the guideline rules and define them more thoroughly (Section 2.2). The pre-defined error types are then identified based on automatic syntactic annotation. For the syntactic annotation, I use state-of-the-art statistical parsers. As discussed in Chapter 1, parsers might need to be adapted to domain-specific texts. To investigate whether "off-the-shelf" parsers are accurate enough to be used for error detection and the degree to which they affect the performance of error detection, I explore various parsers (Section 2.4 and 2.5) after I present the data used for the development and testing of the proposed method (Section 2.3).

2.2 Requirement: Style Rules in Style Guidelines

In this section, I define and refine a set of style rules based on the style guidelines published by the following Swiss governmental bodies:

(1)
1. the Swiss federal administration (Bundesamt für Justiz, 2007)
2. the state administration of the Swiss canton of Bern (Regierungsrat des Kantons Bern, 2000)
3. the state administration of the Swiss canton of Zurich (Regierungsrat des Kantons Zürich, 2005)

In this thesis, I focus on seven syntax-related style rules that are included in these style guidelines as presented in Table 2.1. The rules consist of two types: the first group is purely syntactic (cf. #1 in Table 2.1); the second group concerns syntactic structures related to discourse structures (cf. #2 in Table 2.1). Table 2.1 also provides an overview of the style guides that list style rules.

#	Style rules	CH	BE	ZH
1	Syntactic complexity			
	(1A) Avoid passive constructions	√	√	√
	(1B) Avoid non-canonical word orders	-	-	√
	(1C) Avoid coordinated sentences	√	-	-
	(1D) Avoid embedded clauses	-	√	√
	(1E) Avoid complex noun phrases	-	√	√
2	Discourse structure			
	(2A) Legal conditions should precede legal consequences	√	-	-
	(2B) Legal conditions should not be placed separately in a sentence	-	-	√

TABLE 2.1: Syntax-related style rules in Swiss legislative guidelines (CH = the Swiss federal administration, BE = Bern, ZH = Zurich)

In style guidelines, writing rules are often concrete on orthographic regulations (e.g. abbreviations) and document structures (e.g. title numbering), whereas syntactic rules are relatively simplistic. For example, the Zurich guidelines state that participle phrases embedded in noun phrases should be avoided (Regierungsrat des Kantons Zürich, 2005, pp. 74). Bad and better examples accompany this rule (2). Accordingly, a relative clause (2-b) is better than a participle phrase (2-a). However, this type of guideline is not specific enough to define style rules in rule-based error detection. In order to implement this detection method, we need to know the threshold of tolerance for syntactic complexity. For example, would a noun phrase such as *der im Kanton bekannten definitiven Steuerfaktoren* be less complex than (2-a)? And would it be still considered to be a style error? How many embedded participle phrases in a noun phrase are acceptable?

(2) a. Bad example:
Die Berechnung erfolgt auf Grund *der am 1. Januar des dem Auszahlungsjahr vorangehenden Jahres im Kanton bekannten definitiven Steuerfaktoren.*

b. Better example:
Die Berechnung erfolgt auf Grund der Steuerfaktoren. Massgebend sind *die letzten Faktoren, die am 1. Januar des dem Auszahlungsjahr vorangehenden Jahres definitiv vorliegen.*

In this section, I describe in detail the seven style rules in the style guidelines presented in Table 2.1. They are then concretized for operationalization. I then provide an overview of the linguistic annotation required for the automatic detection of the defined error types.

2.2.1 Syntactic Complexity

Passive Constructions The frequent use of passive constructions indicates that texts are abstract and technical in content and formal in style (Biber, 1988, pp. 112). Because the agent is often not relevant from a legal perspective, law texts tend to overuse agentless, passive constructions (Nussbaumer, 2009, pp. 2,138). However, the style guidelines unanimously state that the passive form is generally to be avoided (Bundesamt für Justiz, 2007, pp. 383f; Regierungsrat des Kantons Zürich, 2005, pp. 74; Regierungsrat des Kantons Bern, 2000, pp. 10). The Bern style guide gives a bad example and a better example of the use of the passive voice (3). The Zurich guidelines state that the passive form is periphrastic, which makes it difficult to read, because all the passage must be read before the verb is revealed at the end of a clause (Regierungsrat des Kantons Zürich, 2005, pp. 73). Nonetheless, the style guidelines also suggest that passive sentences are not always stylistically wrong. For example, the Bern style guideline states that passive sentences are not avoidable if agents are (i) more than one person and are not definable or (ii) should not be mentioned or are not relevant enough to be mentioned or cannot be specified or if (iii) the passive form is better for the realization of a topic–comment structure (Regierungsrat des Kantons Bern, 2000, pp. 10). In style checking tools, the passive form is prototypically regarded as a style error (e.g. Reuther, 1998). In some systems, such as the UNIX Writer's Workbench (MacDonald, 1983), the passive form is flagged only if it exceeds a certain frequency.

(3) a. Bad example:
Die Gesuche werden vom Amt innert 20 Tagen geprüft.

b. Better examples:

(i) Das Amt prüft die Gesuche innert 20 Tagen.

(ii) Die Gesuche werden innert 20 Tagen geprüft.

In the method of automatic error detection developed in this thesis, I define the *werden*-passive construction as erroneous only if the agent is overt. In this case, passive constructions can be rephrased into active constructions.

Canonical Word Orders Word order is relatively free in German although there are canonical word orders (Lenerz, 1977). The Zurich style guide states that word order should be canonical, providing good and bad examples (4) (Regierungsrat des Kantons Zürich, 2005, pp. 73). Several style checking tools flag word orders that are not canonical (e.g. Fortmann and Forst, 2004, Reuther, 1998, Thurmair, 1990).

(4) a. Bad example:
Die Bewilligung erteilt das Amt.
b. Better example:
Das Amt erteilt die Bewilligung.

In the automatic error detection developed in this thesis, noncanonical word orders are concretized as (5), and they are flagged regardless of clause types.

(5) a. Pronouns follow full nouns if they are both arguments.
b. Subject follows direct object or indirect object.
c. Pronominal direct object follows pronominal indirect object.

Accordingly, the following three sentences are flagged due to violation of the respective rules (5-a), (5-b) and (5-c):

(6) a. Wenn die Wahrung der Interessen des Landes es erfordert, kann der Bundesrat Verordnungen und Verfügungen erlassen.[1]
b. Dem Kanton stehen das Salzregal, das Bergregal und das Verfügungsrecht über das Grundwasser, den Gemeinden das Jagd- und das Fischereiregal zu.
c. Erfüllt die antragstellende Person die Voraussetzungen nicht, so eröffnet die Zollverwaltung ihr dies mit einem Nichteintretensentscheid.

Coordinated Sentences In a coordinated sentence, two or more independent clauses are linked by a coordinating conjunction. The coordinated sentence is a linguistic indicator of the violation of Eugen Huber's rules (Höfler, 2011): According to Eugen Huber, the creator of the Swiss Civil Code, an article should not contain more than three paragraphs, a paragraph should consist of only one sentence, and a sentence should express only one idea (norm) (Bundesamt für Justiz, 2007, pp. 359). This rule guides the organization of legal content into articles, paragraphs, and sentences. Coordinated sentences

[1] Examples in this chapter stem from Swiss German-language law texts (cf. Section 1.5). If I modify and rephrase original law texts, I mark them respectively.

are among the typical syntactic structures that contain more than one norm in a sentence. In the example (7), the sentence describes two duties of the Federal Office: the coordination of federal statistical offices, and the creation of standards. Coordinated sentences could be improved by rephrasing them as enumerated structures, such as *The Federal Office has the following duties: (a) ... (b)....*

(7) Das Bundesamt koordiniert die Bundesstatistik und erstellt einheitliche Grundlagen im Interesse der nationalen und internationalen Vergleichbarkeit.

In the automatic detection method developed here, I define coordinated sentences as stylistically incorrect if they are main clauses and if they are S-coordinated or VP-coordinated. Coordinated sentences are not regarded as stylistically incorrect if they are embedded in enumeration structures. Enumeration structures indicate that their listed elements belong to the same topic. Coordinated sentences in subordinated clauses are not flagged.

Embedded Clauses In linguistics, as well as in style checking tools, embedded clauses are regarded as difficult syntactic structures (e.g. Gibson, 2000, Reuther, 1998, Chomsky, 1965). The example (8) shows a typical center-embedded clause. A relative clause *die bei ihm ...* is embedded in a free relative clause *Wer als Depotvertreter ....*

(8) Wer als Depotvertreter Mitwirkungsrechte aus Aktien, die bei ihm hinterlegt sind, ausüben will, ersucht den Hinterleger vor jeder Generalversammlung um Weisungen für die Stimmabgabe.

In the automatic detection method developed here, I define embedded clauses as a style violation if subordinate clauses, relative clauses, and nonfinite clauses are mutually embedded or are inserted in the middle of the vorfeld or the mittelfeld in main clauses. Vorfeld and mittelfeld are sub-fields in the topological field model (Höhle, 1986) which I will discuss in detail in Section 3.2.1.

Complex Noun Phrases German complex noun phrases are similar to reduced relative clauses in English, but they embed participle phrases. Complex noun phrases are treated as difficult-to-read in style checking tools (e.g. Reuther, 1998). Style guidelines suggest that complex noun phrases can be rephrased as relative clauses or prepositional phrases. The Zurich and Bern style guidelines provide bad and better examples of the sentence pair (9) (the sentences are the same as (2)) and (10), respectively.

(9) a. Bad example:
Die Berechnung erfolgt auf Grund der am 1. Januar des dem Auszahlungsjahr vorangehenden Jahres im Kanton bekannten definitiven Steuerfaktoren.
b. Better example:
Die Berechnung erfolgt auf Grund der Steuerfaktoren. Massgebend sind die letzten Faktoren, die am 1. Januar des dem Auszahlungsjahr vorangehenden Jahres definitiv vorliegen.

(10) a. Bad example:
Jedermann hat Anspruch auf Geheimhaltung der ihn betreffenden personenbezogenen Daten.
b. Better example:
Jedermann hat das Recht auf Geheimhaltung der personenbezogenen Daten, die ihn betreffen.

In the automatic detection method developed here, I define complex noun phrases as stylistically incorrect if they embed another noun phrase, following the example pair (10). Therefore, the sentences (9-a), (9-b) and (10-a) are flagged in the error detection.

2.2.2 Discourse Structures

Prototypically, legal norms are formulated as conditional clauses that are paired with consequences (Nussbaumer, 2009, pp. 2137). Legislative style guidelines propose two ways to structure the discourse relation *condition* syntactically. First, conditions should precede consequences (Bundesamt für Justiz, 2007, pp. 356). This order reflects the sequence of acts of conditions and consequences. In example (11), consequences (*Ein Schweizer ...*) are followed by conditions (*wenn er oder sie ...*), which violates the rule. Second, conditions should be expressed in one place (Regierungsrat des Kantons Zürich, 2005, pp. 72f). However, example (12) does not comply with this rule because two conditions, *Wird ...* and *sofern..*, are positioned syntactically in separate fields.

(11) Ein Schweizer Bürger wird auf Begehren aus dem Bürgerrecht entlassen, wenn er in der Schweiz keinen Wohnsitz hat und eine andere Staatsangehörigkeit besitzt oder ihm eine solche zugesichert ist.

(12) Wird Angehörigen des Polizeikorps, die mindestens den Grad eines Korporals bekleiden, eine Aufgabe zugeteilt, für die im Stellenwertstufenplan ein

> höherer Dienstgrad als Wachtmeister vorgesehen ist, wird ihnen eine Funktionszulage ausgerichtet, sofern die Beförderung in den betreffenden Dienstgrad nicht möglich ist. (Regierungsrat des Kantons Zürich, 2005, pp. 72)

In the automatic error detection method developed here, I define conditional clauses or phrases as violations of the first rule if they are not in the vorfeld. The second rule is violated if they are separated with regard to sub-fields (i.e. vorfeld, mittelfeld, and nachfeld) (cf. Section 3.2.1).

Condition To detect violations of these two style rules automatically, conditional clauses and phrases must be identified. To achieve this task, I use lexico-grammatical patterns (13) based on linguistic analysis (cf. Soffritti, 2009, 2013). Accordingly, conditional clauses and phrases can be identified based on linguistic cues, such as clause types (e.g. verb-first clauses), discourse connectives (e.g. *wenn, falls*), and phrasemes (e.g. *in ... Fall*).

(13)
a. verb-first clauses
b. verb-second clauses with the following prepositional phrases if they are dependent on verbs: *bei, mangels, in ... Fall*
c. verb-final clauses with the following discourse connectives: *wenn, falls, soweit, sobald, solange, sofern, insoweit, insofern, inwieweit, ausser*
d. verb-final clauses with the free relative pronoun *wer*
e. verb-final clauses with the relative pronouns *der, welcher* in vorfeld

These linguistic indicators are often ambiguous in the mapping of discourse relations. Discourse connectives are often associated with more than one discourse relation e.g. *wenn, aber* (cf. Bayerl, 2004), and they are ambiguous between connectives and non-connectives e.g. *als, während, dabei, so* (cf. Bögel et al., 2014, Schneider and Stede, 2012, Dipper and Stede, 2006). Furthermore, verb-first clauses are not always conditional clauses. They also can be interrogative and imperative clauses. Because the domain examined in this thesis is restricted to law texts, the linking of connectives to the discourse relation *condition* is less ambiguous. For example, the connective *wenn* can signal temporal or conditional relations. However, because law texts are normative and contain neither argumentative nor narrative discourses, the connective *wenn* is usually linked to the discourse relation *condition*. Similarly, in law texts, verb-first clauses are associated with conditions. Because law texts exclude inter-personal communication, imperative and interrogative moods rarely occur.

In this section, I concretized a set of syntax-related rules in style guidelines and defined error types of syntactic structures to be detected automatically. In the next section, I provide an overview of the syntactic cues that are required for automatic detection.

2.2.3 Required Syntactic Annotation

Table 2.2 provides an overview of the syntactic cues used in the detection of each erroneous type of syntactic structure that I defined in the previous section. Most require the annotation of clause boundaries (or chunks), clause types (e.g. verb-first clause), and topological fields (i.e. vorfeld, mittelfeld, nachfeld). The error type (1B) also requires information about the syntactic functions of arguments such as the subject.

#	Error types	Linguistic requirement
(1A)	Agentive passive sentences	Agentive *werden*-passive construction with overt agent in a clause
(1B)	Marked word orders	Linear order of arguments in a clause
(1C)	Coordinated sentences	Coordinated main clause structure
(1D)	Embedded clauses	Embedded clause structure
(1E)	Complex noun phrases	Noun phrases that embed another noun phrase whose head is an attributive present participle, past participle, zu-infinitive, or adjective
(2A)	Consequences before conditions	Conditional clause and phrase in mittelfeld (verb-final subordinated clause with a conditional connective, conditional prepositional phrase, free relative clause)
(2B)	Separated conditions	Conditional clause and phrase in vorfeld (verb-first clause, verb-final subordinated clause with a conditional connective, conditional prepositional phrase, relative clause, free relative clause) and those in mittelfeld (cf. (2A))

TABLE 2.2: Linguistic requirement for the automatic detection of the error types

In this section, I presented a set of problematic syntactic structures based on legislative style guidelines and the linguistic requirements for their automatic detection. In the following sections, I describe the implementation of the rule-based error detection method developed here. Next, I describe the data used for the development and evaluation of the error detection system.

2.3 Data: Building a Test Suite

I built a test suite in order to develop and test the automatic error detection system. The test suite is a collection of sentences containing each of the seven types of violations

and non-violations illustrated in Table 2.1.

Using Swiss German-language law texts (cf. Section 1.5), I semi-automatically collected erroneous (i.e. stylistically incorrect) and near erroneous (i.e. stylistically correct) sentences containing each type of error. In the first step of the data collection, sentences that were likely to violate a style rule were automatically extracted based on word form and on the lemma and POS tag generated by TreeTagger (Schmid, 1999). Table 2.3 provides an overview of the search terms. They were defined as generally as possible, i.e. recall-oriented, so that the test suite contained a broad coverage of syntactic variations per type. I then manually examined the automatically extracted structures[2] and collected the instances into a test set and a development set. For each error type, the test set contained 50 items for style violations and 50 items for near style violations. In addition, I collected a few separate items of each error type in order to develop the error detection system.

[2] In this task, a legislative editor helped to judge the style of syntactic structures.

#	Errors types and search cues
(1A)	**Agentive passive sentence:** A sentence is extracted if it contains the auxiliary verb *werden*, a participle past and a preposition (*von* or *durch*).
(1B)	**Marked word orders:** A sentence is extracted if: 1. a pronoun is followed by a full noun 2. a nominative pronoun, or a determiner is followed by a dative or accusative pronoun or determiner.
(1C)	**Coordinated sentence:** A sentence is extracted if a finite verb follows a coordinating conjunction.
(1D)	**Embedded clauses:** A sentence is extracted if there are more than one subordinating conjunction, nonfinite conjunction or relative pronoun.
(1E)	**Complex noun phrase:** A sentence is extracted if: 1. a determiner is adjacent to a determiner, preposition, or pronoun or 2. a preposition or comparative conjunction is not adjacent to a determiner, adjective, noun, or pronoun.
(2A)	**Sequences before conditions** A sentence is extracted if it contains a conditional discourse connective (e.g. *falls*), relative pronoun (e.g. *wer*), or conditional preposition (e.g. *bei*) that precedes a finite verb.
(2B)	**Separated conditions** A sentence is extracted if it contains more than one conditional discourse connective, relative pronoun, or conditional preposition.

TABLE 2.3: Extraction of candidate sentences for the creation of the test suite for the error detection

2.4 Baseline: Use of Dependency Grammar Parsers

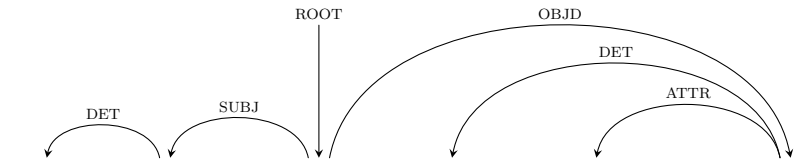
Error detection consists of two sequential steps: 1) automatic syntactic annotation, i.e. parsing; and 2) detection of style violations based on 1). The first task provides the linguistic information required by the second task. As I discussed in section 2.2.3, the required linguistic information includes clause segmentation, clause types, topological fields, and syntactic functions. In this chapter, I use a dependency grammar parser for the automatic linguistic annotation. In the dependency grammar (Foth, 2005), clause structures and syntactic functions are represented as dependency relations. In the next section, I briefly summarize the dependency grammar.

2.4.1 Dependency Grammar

A dependency grammar (Tesnière, 1959) is a word grammar in which words are connected by grammatical relations. The relations between two words are asymmetric. Two words are in a dependency relation in which one is the head i.e. the governor, and the other is the dependent. For example, the German sentence (14) is analyzed in dependency grammar as (15) (Foth, 2005).

(14) Dieses Gesetz untersteht dem fakultativen Referendum.

(15)



Asymmetric relations are represented by directed paths from heads to dependents. Dependency grammar relations are assigned to paths. In dependency grammar, the predicate of a clause is the head, and the head nouns of syntactic functions (here: subject `SUBJ`, dative object `OBJD`) are the dependents, which provide the core meaning of a clause ("who does what"). The head nouns are again structured by having dependent attributives (`ATTR`) and determiners (`DET`). The `ROOT` is the only dependency node in a sentence that is not dependent on any nodes and is the ancestor of all words in the sentence. Dependency grammar relations often are described as family relations, as shown in (16).

(16)
a. The dependency relation in the first order as mother and child (e.g. *untersteht* and *Gesetz* in the sentence (14))
b. The dependency relation in the second order as grandmother and grandchild (e.g. *untersteht* and *Dieses* in the sentence (14))
c. The dependency relation in the nth order as ancestor and descendent.

2.4.2 Implementation

For dependency analysis, I used the state-of-the-art statistical graph-based dependency parser developed by Bohnet (2010). The Bohnet parser[3] was the best parser for the labeling task in German in the CoNLL-2009 Shared Task.[4] I trained the Bohnet parser by using 60% of TüBa-D/Z (700,888 tokens) (cf. Section 1.5).[5]

For the detection task, I used Constraint Grammar (CG) (Karlsson et al., 1995) as the framework for creating the detection rules. The CG compiler searches targeted error types based on the condition of the linear order of the tokens or the dependency paths of the tokens in a sentence.[6] Table 2.4 provides an overview of the detection rules in dependency grammar format. I created the rules based on the error types described in Table 2.2.

[3]Bohnet parser is available at: http://code.google.com/p/mate-tools/ (last visited on 01/06/2015). I used graph-based version anna-3.6 of the parser and trained the data with first, second, and third order factors.

[4]Information is available at: http://ufal.mff.cuni.cz/conll2009-st/index.html (last visited on 01/06/2015)

[5]The training data is the same as 'TüBa60Train' described in Chapter 5. In the Chapter, 60% of TüBa is used for the training, and 40% for the cross validation and evaluation to develop a statistical supertagger.

[6]I used VISL3 as CG compiler: VISL3 is available at http://beta.visl.sdu.dk/constraint_grammar.html (last visited: 18/6/2015)

#	Targeted error types and dependency grammar relations
(1A)	**Agentive passive sentences:** A lemma *werden* is marked if the sentence contains an overt agent and a past participle. The agent is a noun of which the mother is the lemma *von* or *durch* (`PP`) that is attached to verb.
(1B)	**Marked word orders:** The linear order of the syntactic functions of arguments in a clause and the pronominality of nouns are used to identify marked word orders (Clause boundaries are recognized based on dependency grammar relations and POS tags).
(1C)	**Coordinated sentences:** A coordinated token, `KON` or `CJ`, is marked if it is a finite verb and the descendent of a `ROOT`. Enumeration structures are excluded.
(1D)	**Embedded clauses:** A relative pronoun, subordinating conjunction, complementizer or nonfinite conjunction is marked: 1. if it is embedded in a subordinated clause (i.e. if the parent is the ancestor of `NEB`, `REL`, `OBJI`, `OBJC` or `SUBJC` and precedes it.) 2. if it is placed in the middle of a vorfeld. (i.e. if the parent precedes `NEB`, `REL`, `OBJI`, `OBJC` or `SUBJC` that is followed by main verb `ROOT`), or 3. if it is placed in the middle of a mittelfeld (i.e. if the parent precedes `ROOT` and follows `AUX`).
(1E)	**Complex noun phrases:** A determiner is marked if it is the child of the head noun of a participle phrase (i.e. the head noun has an adjective or a participle as the child, and a pronoun, noun or preposition as the grandchild. A preposition is marked if it is the mother of the head noun of a participle phrase.
(2A)	**Sequences before conditions** Conditional phrases or clauses are marked if they are in the mittelfeld (i.e. it follows `ROOT` and precedes `AUX` that is the child of `ROOT`). Conditional clauses and phrases in the mittelfeld are: 1. adverbial subordinated clauses (`NEB`) with a conditional connective (e.g. *wenn, falls*) 2. free relative clauses (`SUBJC/OBJC`) with *wer*, or 3. verb-attached adjunct conditional prepositional phrases (`PP`) (e.g. *bei, mangels, in ... Fall*). Enumeration structures are excluded.
(2B)	**Separated conditions** Conditional clauses and phrases are marked if they are placed in the vorfeld and in the mittelfeld. Conditional clauses and phrases in the vorfeld are: 1. verb-first clauses (`NEB`) 2. verb-final subordinated clauses (`NEB`) with a conditional connective 3. free relative clauses (`SUBJC/OBJC`) with *wer* 4. relative clauses (`REL`), or 5. verb-attached adjunct conditional prepositional phrases (`PP`) Conditional clauses and phrases in the mittelfeld are the same as in (2A). Enumeration structures are excluded.

Table 2.4: Error detection rules for the dependency grammar

2.4.3 Evaluation

Test I tested the detection of erroneous syntactic structures on the test suite presented in Section 2.3. I then evaluated the performance in terms of precision, recall, and the F1 score. Hereby, precision estimates how many percent of the detected style violations are actually style errors. Recall estimates how many percent of style violations are actually retrieved from the test set. The F1 score harmonizes the percentages of precision and recall (details are provided in Chapter 4).

Table 2.5 shows an overview of the results. As the table shows, precision was higher than recall at scores of 88.06 and 50.57, respectively. The combined results for precision and recall yielded an F1 score of 64.26. The detection of style guide violations concerning complex noun phrases and agentive passive sentences were high at F1 scores: 90.32 and 89.58, respectively.

(#) Target	Precision	(retrieved #)	Recall	(total #)	F1
(1A) Agentive passive sentences	93.48	(46)	86.00	(50)	89.58
(1B) Marked word orders	100.00	(9)	18.00	(50)	30.51
(1C) Coordinated sentences	84.09	(44)	74.00	(50)	78.72
(1D) Embedded clauses	77.78	(9)	14.00	(50)	23.73
(1E) Complex noun phrases	97.67	(43)	84.00	(50)	90.32
(2A) Consequences before conditions	58.33	(12)	14.00	(50)	22.58
(2B) Separated conditions	84.21	(38)	64.00	(50)	72.73
Total	88.06	(201)	50.57	(350)	64.26

TABLE 2.5: Evaluation of the error detection system using the dependency parser

Error Analysis Because precision is important in error detection tasks, I conducted an error analysis of false positives (i.e. correct sentences that were detected as style violations) to gain insight into the errors made by the developed method. Table 2.6 provides an overview of the results. Half of the system errors are caused by the parser (14 errors) and less than half by the error detection method (11 errors).

(#) Target	Preprocessing errors	Parsing errors	Detection errors
(1A) Agentive passive sentences	1	1	1
(1B) Marked word orders	0	0	0
(1C) Coordinated sentences	0	8	0
(1D) Embedded clauses	1	2	0
(1E) Complex noun phrases	1	0	0
(2A) Consequences before conditions	0	3	3
(2B) Separated conditions	0	0	6
Total	2	14	11

TABLE 2.6: Error analysis of false positives: error detection system using the dependency parser

The parsing mistakes mainly originated in the dependency grammar relations with regard to clause recognition (9 of 14 errors). In particular, the labeling of `ROOT` was problematic. For instance, 6 out of 9 label errors in the error type (1C) involved `ROOT`. The parsed sentences (17) and (18) illustrate a typical example of this problem. In (17), coordinated verb-first clauses were not recognized by the parser: The word *verweigert* was labeled as `ROOT` instead of coordination of *Nimmt* i.e. `KON`. In the sentence (18), *annimmt* is correctly labeled as `CJ`, but incorrectly linked to `ROOT` *sind* instead of `NEB` *beantragt*. These parser errors led to detection errors.

(17) Nimmt die für den Kulturgüterschutz zuständige Stelle des Bundes bei der Beitragszusicherung Kürzungen vor, *verweigert*_`ROOT` sie die Beiträge oder *nimmt*_`CJ` sie bei der Revision von Abrechnungen Kürzungen vor, so muss sie dies begründen.

(18) Diese Angaben *sind*_`ROOT` dem Versicherungsnehmer so zu übergeben, dass er sie kennen kann, wenn er den Versicherungsvertrag beantragt oder *annimmt*_`CJ`.

During the detection process, a minority of the system errors were related to the under- and over-specification of detection rules. The mistakes made in the style error detection were mostly in connection with discourse recognition errors (syntax-discourse rules (2A) and (2B)). 7 out of 11 detection errors were due to this type of error. In (2A) and (2B), the recognition of conditional phrases, particularly those with a preposition *bei*, was problematic. The preposition *bei* can be conditional (cf. (19-a)), temporal (cf. (19-b)), or local (cf. (19-c)). This syntactic cue was too ambiguous to be directly mapped to the discourse relation *condition*.

(19) a. *Conditional*
Stauanlagen müssen **bei** drohender Gefahr abgesenkt und zur Vornahme von Kontroll- und Unterhaltsarbeiten entleert werden können.

b. *Temporal*
Wird ein Entzug des Lernfahrausweises oder des Führerausweises auf unbestimmte Zeit verfügt, so teilt das BAV der betroffenen Person **bei** der Eröffnung der Verfügung die Bedingungen für die Wiedererteilung des Lernfahrausweises oder des Führerausweises mit.

c. *Local*
Betrifft eine Verfügung jedoch Ansprüche vermögensrechtlicher Art, so kann **bei** der Gruppe Verteidigung des VBS dagegen Beschwerde erhoben werden.

In summary, the results showed that the rule-based error detection system based on the dependency parser was accurate for the detection of style violations of agentive passive constructions and complex noun phrases. The evaluation showed that detection errors were mainly caused by parsing errors and discourse detection errors. In the next section, I try to improve the accuracy of the error detection system by optimizing the parser for the error detection task.

2.5 Improvement: Use of Topological Dependency Grammar Parsers

To reduce parser errors in the error detection system, I propose a syntactic representation, 'topological dependency grammar', for the syntactic annotation. The previous section showed that most parsing errors were caused by clause recognition. The aim is to replace problematic clause-boundary dependency labels, such as `ROOT` and `NEB`, by less semantic-oriented and rather linear-order-oriented topological field relations.

In the next section, I present the topological dependency grammar in detail. I then revise and re-test the implementation of the error detection rules using the topological dependency grammar.

2.5.1 Topological Dependency Grammar

The proposed topological dependency grammar is a syntactic tag set comprised of two types of grammars: dependency grammar (Foth, 2005) and topological fields (Höhle,

1986). In the tag set, I introduce topological fields for the representation of clause structures. Tokens that belong to clause brackets are analyzed in the model of topological fields.[7] Otherwise, the tokens are analyzed as dependency grammar relations. In this topological dependency grammar, syntactic structures are analyzed linearly and functionally according to clause boundaries, with the objective of minimally annotating texts for the error detection.[8]

A topological dependency grammar analyzer, for example, annotates the sentence (20) as (21). The main verb *untersteht* is analyzed as the left bracket (`LB`) of the verb-second (`V2`) clause. The head noun of the subject *Gesetz* is tagged as `SUBJ`, and the head noun of the indirect object *Referendum* is tagged as `OBJD`. Determiners and attributes are tagged as `DET` and `ATTR`, respectively. Here, all tokens are assigned to dependency grammar tags (i.e. `DET, SUBJ, ATTR, OBJD`), with the exception of `LB-V2`. The left or right brackets of clauses are analyzed according to the concept of topological fields.

(20) Dieses Gesetz untersteht dem fakultativen Referendum.

(21) Dieses_`DET` Gesetz_`SUBJ` untersteht_`LB-V2` dem_`DET` fakultativen_`ATTR` Referendum_`OBJD`.

The following paragraphs briefly describe topological fields and the tag set of dependency grammar relations used in the topological dependency grammar.

Topological Fields The topological field (Höhle, 1986, Drach, 1963) is a traditional grammar model that is used for the linear analysis of German clauses and the placement of verbal elements (Dürscheid, 2012, Sternefeld, 2006). The segmented fields—vorfeld, mittelfeld and nachfeld—are syntactic 'sub-fields' that capture syntactic phenomena such as topicalization and extraposition. In German, the vorfeld is a topic position, and

[7] Clause brackets are, in particular, associated with verbs. In dependency grammar (Foth, 2005), the dependency labels of finite verbs are categorized into `SUBJC/OBJC` for complement clauses and free relative clauses, `NEB` for adjunct clauses (i.e. adverbial sub-clauses or verb-first conditionals), `REL` for relative clauses, `KON/CJ` for coordinated verbs, and `ROOT` for verb-second main clauses or verb-first imperative and interrogative clauses. Thus, the dependency grammar tags of verbs are functionally and semantically oriented.

[8] The idea of combining dependency grammar and the topological field model is not novel. Duchier and Debusmann (2001) proposed a topological dependency grammar that consisted of two distinct trees: a topological tree and a dependency tree. Unlike the topological dependency grammar proposed by Duchier and Debusmann (2001), I propose a supertag set consisting of two mutually exclusive grammar categories. Rather, the proposed topological dependency tag set is similar to a constraint grammar parser of English (Voutilainen, 1996, Voutilainen and Tapanainen, 1993). The syntactic tag is based on a dependency-oriented functional grammar with sentence boundaries and types (finite and nonfinite clauses).

it is typically occupied by a subject. In the mittelfeld, the linear order of grammatical functions tends to be canonical. Heavy phrases, typically nonfinite clauses, are located in the nachfeld (Dürscheid, 2012, pp. 95ff).

In the concept of topological fields, sentences are segmented by verbs, placed in left brackets (LB) and right brackets (RB), respectively:

(22) `<Vorfeld>|`*`LB`*`|<Mittelfeld>|`*`RB`*`|<Nachfeld>`

Therefore, the recognition of topological fields corresponds to that of the clause brackets, LB and RB. Topological sub-fields are defined relative to the left and right brackets. The vorfeld is the sub-field positioned before LB. The mittelfeld is the sub-field between LB and RB, and the nachfeld is the sub-field after RB.

Depending on the distribution of the vorfeld and the clause brackets (LB and RB), three types of German clauses are distinguished: verb-first (V1), verb-second (V2), and verb-final (VF) clauses (cf. Table 2.7). V1 clauses are imperative sentences, interrogative sentences, and certain conditional clauses. V2 clauses are main clauses. VF clauses are subordinated clauses and relative clauses. The vorfeld is occupied by a constituent in a V2 clause, whereas it remains empty in V1 and VF clauses.[9] In V1 and V2 clauses, the left bracket is filled by a finite verb, and the right bracket is optionally filled by a nonfinite verb, a separated verb prefix, or a predicative adjective. In contrast, in a VF clause, the LB is marked by a functional word, such as a subordinating conjunction, complementizer, or relative pronoun. The RB is marked by the verbal complex.

	Vorfeld	**LB**	**RB**
V1	empty	a finite verb	nonfinite verbs or predicative adjectives (optional)
V2	a constituent	a finite verb	nonfinite verbs or predicative adjectives (optional)
VF	empty	a functional word	a finite verb (obligatory) and nonfinite verbs (optional)

Table 2.7: Occupation of vorfeld, left bracket (LB) and right bracket (RB) in V1, V2 and VF clauses

Table 2.8 illustrates how the following example sentence is analyzed by clause types:

[9] In linguistics, relative pronouns are considered to be placed in the vorfeld and not in the left bracket. For the automatic annotation of topological fields, relative pronouns are regarded as left brackets, so no empty tokens need to be tagged.

(23) Stellt die Zollverwaltung Unregelmässigkeiten fest, so verweigert sie den Abschluss des Transitverfahrens und hält die Sicherheit zurück, bis die mit bedingter Zahlungspflicht veranlagten Einfuhrzollabgaben bezahlt sind.

Vorfeld	Left Bracket (LB)	Mittelfeld	Right Bracket (RB)	Nachfeld
Verb-first clause (V1):		LB = finite verb, RB = nonfinite verb(s) or/and a predicative adjective, or a separated verb prefix		
	Stellt	*die Zollverwaltung Unregelmässigkeiten*	*fest,*	
Verb-second clause (V2):		LB = finite verb, RB = nonfinite verb(s) or/and a predicative adjective, or a separated verb prefix		
so	*verweigert*	*sie den Abschluss des Transitverfahrens*		
[und]	*hält*	*die Sicherheit*	*zurück*	
Verb-final clause (VF):		LB = subord. conjunction/ complementizer, RB = verb complex		
	bis	*die mit bedingter Zahlungspflicht veranlagten Einfuhrzollabgaben*	*bezahlt sind.*	

TABLE 2.8: Exemplification of the topological field model: occupation of LB and RB in sentence (23).

Based on the previous description, I propose the following topological field tag set (cf. Table 2.9), which is a combination of types of clause brackets, that is, LB and RB and clause types (`V1`, `V2`, `VF`). Clause brackets can either remain empty (∅ in Table 2.9) or be occupied by restricted morphological categories.[10] Compared to the standard topological annotation, TüBa-D/Z annotation (Telljohann et al., 2012), the topological field tag set is not tree-based but linear-order-based. Therefore, the three German clause types are added to the tag set.[11]

[10] In Table 2.9, finite verbs are `FIN`, nonfinite verbs are `NONFIN` (including predicative adjectives), subordinating conjunctions are `CONJ`, complementizers are `COMP`, relative pronouns are `RC-PRON`, infinite conjunctions are `ICONJ`, punctuation is `PUNCT`, and zu-infinitive verbs are `ZU-INF`. Dependency grammar categories, i.e. subject, objects, determiners and the dependents of prepositions are `DEP`.

[11] TüBa annotation set is de facto the state-of-the art tag set. Based on the annotation, topological field parsers/chunkers were developed (e.g. Cheung and Penn, 2009, Veenstra et al., 2002)

Topological Field Tag	**Clause brackets**	**Morphological Category**
LB-V1	Left bracket of a verb-first clause	FIN
RB-V1	Right bracket of a verb-second clause	NONFIN/∅
LB-V2	Left bracket of a verb-second clause	FIN
RB-V2	Right bracket of a verb-second clause	NONFIN/∅
LB-VF-CONJ	Left bracket of a verb-final clause (subordinated clause)	CONJ/COMP
RB-VF-CONJ	Right bracket of a verb-final clause (subordinated clause)	FIN
DEP=LB-VF-RC	Left bracket of a verb-final clause (relative clause)	RC-PRON
RB-VF-RC	Right bracket of a verb-final clause (relative clause)	FIN
VC	Part of a verbal complex	NONFIN
TOP	Topicalized right-most verb in a verb-second clause	NONFIN
LXCOMP	Left bracket of an infinitive clause	ICONJ/PUNCT
RXCOMP	Right bracket of an infinitive clause	ZU-INF

TABLE 2.9: Topological field tag set

In the tag set, right brackets (i.e. verbal complex) are differentiated in two tags, i.e. RB and VC, to facilitate clause chunking.[12] The right-most verbs are assigned to RB, and the remaining verbs are assigned to VC. RB is assigned to nonfinite verbs in V1 or V2 clauses or to finite verbs in VF clauses. VC is assigned to nonfinite verbs, regardless of the clause type. For example, *bezahlt* and *sind* in the sentence (23) are analyzed as VC, RB-VF-CONJ, respectively:

(24) Stellt_LB-V1 die Zollverwaltung Unregelmässigkeiten fest_RB-V1, so verweigert _LB-V2 sie den Abschluss des Transitverfahrens und hält_LB-V2 die Sicherheit zurück_RB-V2, bis_LB-VF-CONJ die mit bedingter Zahlungspflicht veranlagten Einfuhrzollabgaben **bezahlt**_VC **sind**_RB-VF-CONJ.

In V2 clauses, verb phrases can be topicalized and placed in the vorfeld. In this case, the verbs are placed at the *left* side of the LB. To differentiate it from the *right* bracket, it is marked as TOP:

(25) **Ausgenommen**_TOP bleiben_LB-V2 die in die Autonomie der teilunterstellten Institutionen fallenden Bereiche.

In addition, a relative pronoun is tagged with a topological field tag and a dependency grammar tag (cf. DEP=LB-VF-RC in Table 2.9). A relative pronoun is the left bracket

[12] Different fonts are used to differentiate the tag and annotation RB/VC and linguistic concept RB.

of a relative clause (i.e. LB-VF-RC) and simultaneously bears a syntactic function. For example, a subject relative pronoun is analyzed as SUBJ=LB-VF-RC. The dependency tags that can be combined with LB-VF-RC are SUBJ, OBJA, OBJD, OBJG, OBJA2, DET and PN.

Nonfinite clauses are further categorized as XCOMP. In particular, nonfinite clauses are often extraposed and placed in the nachfeld. In a nonfinite clause, the left-most token (LXCOMP) is occupied by an infinitive conjunction, such as *ohne* or by a punctuation mark. The right-most token (RXCOMP) is occupied by a zu-infinite verb.

(26) Wer_LB-VF-RC auf Waren oder Geschäftspapieren auf Designschutz hinweist _RB-VF-RC, ohne_LXCOMP die Nummer des Designrechts zu_VC nennen_RXCOMP, ist_LB-V2 verpflichtet_RB-V2 ,_LXCOMP die Nummer auf Anfrage unentgeltlich bekannt zu_VC geben_RXCOMP.

Dependency Grammar The 22 dependency grammar tags used in the topological dependency grammar are illustrated in Table 2.10. These tags are the dependency relations assigned to nouns, prepositions, determiners, attributive adjectives, adverbs, and some conjunctions. The tag set does not include clause-boundary dependency relations that correspond to topological field tags (cf. Table 2.9). The 22 tags are based on the dependency grammar set proposed by Foth (2005).[13]

[13]Foth's dependency tag set is the standard tag set in German. Based on Foth's tag set, Telljohann et al. (2012) built a syntactically annotated German newspaper corpus, TüBa-D/Z, and Sennrich et al. (2013, 2009) developed a German dependency parser.

Dependency tags	**Description**	**Syntactic Category**
SUBJ	Subject	Noun
OBJA	Accusative (direct) object	Noun
OBJD	Dative (indirect) object	Noun
OBJG	Genitive object	Noun
OBJA2	Second accusative (direct) object	Noun
NP2	Subject in an ellipsis coordination structure	Noun
EXPL	Expletive *es*	Noun
GMOD	Genitive modification	Noun
PN	Dependent of a preposition	Noun
APP	Apposition	Noun
ZEIT	Time expression	Noun
GRAD	Grade expression	Noun
PRED	Predicate	Noun, preposition, (pred.) adjective
PP	Adjunct preposition	Preposition
OBJP	Complement preposition	Preposition
DET	Determiner	Determiner
ATTR	Attributive	Adjective
ADV	Adverb	Adverb
KOM	Comparative conjunction	Conjunction
PART	Particle	Particle
KON	Coordination (with comma)	Noun, preposition, determiner, adjective, adverb, conjunction
CJ	Coordination (with conjunction)	Noun, preposition, determiner, adjective, adverb, conjunction

Table 2.10: Dependency grammar tag set for the topological dependency grammar

2.5.2 Implementation

For topological dependency parsing, I re-trained the Bohnet parser for the same TüBa-D/Z training data as used with the dependency Bohnet parser in the previous section. The training data were automatically converted from dependency grammar tags to topological dependency grammar tags (cf. Appendix D).[14]

[14]The training data are the same as those used for the rule-based supertagger (`TüBa60TrainingTopo`) in Section 4.2.4.2.

For error detection, I manually transformed the detection rules used for the dependency grammar into those for the topological dependency grammar labels (cf. Table 2.11). Unlike the detection rules using dependency grammar, the rules are solely based on the conditions of the linear order of tokens.

#	Targeted error types and topological field tags
(1A)	**Agentive passive sentences:** A lemma *werden* is marked if there is a past participle and an overt agent in a clause. The agent is in the prepositional phrase with the preposition *von* or *durch*. The preposition is adjunct (`PP`) and not adjacent to a noun (because of the attachment ambiguity).
(1B)	**Marked word orders:** The linear order of the syntactic functions of the arguments in a clause and the pronominality of nouns are used to identify marked word orders (Clause boundaries are clause brackets).
(1C)	**Coordinated sentences:** In a coordinated sentence, there is more than one left bracket of a verb-second clause (`LB-V2`). Enumeration structures are excluded.
(1D)	**Embedded clauses:** The left bracket of a verb-final clause (`LB-VF-CONJ/RC`) or nonfinite clause (`LXCOMP`) (i.e. relative pronoun, subordinating conjunction, complementizers or nonfinite conjunction) is marked: 1. if the clause is embedded in another subordinated clause (i.e. if it is followed by another left bracket without being interrupted by any clause brackets) 2. if it is placed in the middle of the vorfeld (i.e. if it precedes the left bracket of a verb-second clause (`LB-V2`) and follows the left bracket of a verb-final clause (`LB-VF`) or a verb-first clause (`LB-V1`), or 3. if it is placed in the middle of the mittelfeld (i.e. if it is preceded by the left bracket of a verb-second clause (`LB-V2`) and it is followed by the right bracket of a verb-second clause (`RB-V2`)).
(1E)	**Complex noun phrases:** In a complex noun phrase, a determiner (`DET`) is marked if it is followed by a preposition, adverbial pronoun, pronoun, or comparative conjunction without being interrupted by any tokens, with the exception of attributives (`ATTR`), coordinating conjunctions (`KON/CJ`), and commas. A preposition is marked if it is followed by a preposition without being interrupted by any tokens, with the exception of `ATTR`, `KON`, `CJ` and commas.
(2A)	**Sequences before conditions** A conditional clause or phrase is marked if it is in the mittelfeld (i.e. it follows the left bracket of a verb-second clause (`LB-V2`) and precedes the right bracket of a verb-second clause (`RB-V2`)). A conditional clause or phrase in the mittelfeld is defined as the left bracket of an adverbial subordinated clause (`LB-VF-CONJ`) with a conditional connective, the left bracket of a free relative clause (`LB-VF-RC`), or a preposition (`PP`) with a conditional connective. Enumeration structures are excluded.
(2B)	**Separated conditions** Conditional clauses or phrases are marked if they are placed in the vorfeld (i.e. precede the left bracket of a verb-second clause (`LB-V2`)) and the mittelfeld (cf. (2A)). A conditional clause or phrase in the vorfeld is defined as the left bracket of a verb-first clause (`LB-V1`) or the left bracket of an adverbial subordinated clause (`LB-VF-CONJ`) with a conditional connective, the left bracket of a free/not-free relative clause (`LB-VF-RC`), or a preposition (`PP`) with a conditional connective. Conditional clauses and phrases in the vorfeld are marked the same as in (2A). Enumeration structures are excluded.

TABLE 2.11: Error detection rules for the topological dependency grammar

2.5.3 Evaluation

Test Using the topological dependency parser, I test the error detection method by applying the same test set and evaluation setting as described in the previous section. The results are shown in Table 2.12.

(#) Target	Precision	(retrieved #)	Recall	(total #)	F1
(1A) Agentive passive sentences	89.74	(39)	70.00	(50)	78.65
(1B) Marked word orders	100.00	(10)	**20.00**	(50)	**33.33**
(1C) Coordinated sentences	83.67	(49)	**82.00**	(50)	**82.83**
(1D) Embedded clauses	69.70	(33)	**46.00**	(50)	**55.42**
(1E) Complex noun phrases	85.45	(55)	**94.00**	(50)	89.52
(2A) Consequences before conditions	**88.46**	(52)	**92.00**	(50)	**90.20**
(2B) Separated conditions	**87.80**	(41)	**72.00**	(50)	**79.12**
Total	85.30	(279)	**68.00**	(350)	**75.67**

TABLE 2.12: Evaluation of the error detection system using the topological dependency parser (The improvement is marked with bold font)

The F1 score (75.67% over 64.26%) showed that in the error detection system, the use of the topological field parser outperformed that of the dependency grammar parser. Error detection using the topological dependency parser performed better in 5 of 7 style rules in the F1 score. This improvement was due to the increase in 6 of 7 style rules in recall at a small cost of precision. The results of the error detection method using different parsers further indicate that its accuracy is strongly affected by the preprocessing parser. As the error detection is based on syntactic structures returned by parsers, the results are not surprising, but show the magnitude of the parsing effects on the error detection task. In addition, topological dependency relations are clearly an advantage in the developed error detection system.

Error Analysis I conducted an error analysis of false positives on the test set. Table 2.13 provides an overview of the results. The majority of errors (27 out of 42) were due to parsing errors.

(#) Target	Preprocessing errors	Parsing errors	Detection errors
(1A) Agentive passive sentences	1	2	1
(1B) Marked word orders	0	0	0
(1C) Coordinated sentences	0	8	0
(1D) Embedded clauses	1	9	0
(1E) Complex noun phrases	0	8	0
(2A) Consequences before conditions	0	0	6
(2B) Separated conditions	1	0	5
Total	3	27	12

TABLE 2.13: Error analysis of false positives: error detection system using the topological dependency parser

For rule (1C), false positives were mostly derived from the same type of parsing errors as those made in dependency parsing. For example, *nimmt* in (27) and *annimmt* in (28) were both analyzed as the left bracket of a main sentence (i.e. `LB-V2`). For rule (1D), all mistakes involved the right bracket of a main clause (`RB-V2`): *werden* in (29) and *abgebrochen* in (30). They were analyzed as the right bracket of a main sentence (i.e. `RB-V2`) instead of the right brackets of a verb-final subordinated clause (`RB-VF-CONJ`). For rule (1E), 4 of 8 errors were due to the error of the parser, which returned determiners instead of relative pronouns (cf. (31)). The errors originated from the POS tagger's mistakes. Thus, the parser has difficulties in analyzing syntactically complex sentences. This result indicates that a statistical parser is required to be adopted to the domain, in particular, if the sentences contain complex clause structures.

(27) Nimmt die für den Kulturgüterschutz zuständige Stelle des Bundes bei der Beitragszusicherung Kürzungen vor, verweigert sie die Beiträge oder *nimmt*_`LB-V2` sie bei der Revision von Abrechnungen Kürzungen vor, so muss sie dies begründen.

(28) Diese Angaben sind dem Versicherungsnehmer so zu übergeben, dass er sie kennen kann, wenn er den Versicherungsvertrag beantragt oder **annimmt**_`LB-V2`.

(29) Eine Verteilung darf bereits nach Ablauf von drei Monaten erfolgen, wenn ein zugelassener Revisionsexperte bestätigt, dass die Schulden getilgt sind und nach den Umständen angenommen werden kann, dass keine Interessen Dritter gefährdet **werden**_`LB-V2`.

(30) Die Ausbildung gilt auch als beendet, wenn sie **abgebrochen**_`LB-V2` oder unterbrochen wird oder wenn ein Anspruch auf eine Invalidenrente entsteht.

(31) Das Gemeinwesen, in **dem_DET** der Nutzen einer staatlichen Leistung anfällt, trägt deren Kosten.

In summary, this section showed that the topological dependency representation improved the accuracy of the error detection. By changing the linguistic representation from a hierarchical dependency grammar to flat topological dependency syntactic structures, the F1 score of error detection increased from 64.26% to 75.67%.

2.5.4 Discussion: Domain Difference in Parsing

To gain insights into the effectiveness of the preprocessing parsers, I evaluated the performance of topological dependency parser in analyzing Swiss German-language law texts (400 sentences, 10,330 tokens) and 20% of TüBa-D/Z texts (232,560 tokens) with regard to the labeling of topological dependency relations.[15] I also investigated the degree to which differences in domains affected the performance of a statistical parser (cf. Gildea (2001)).

Table 2.14 shows the accuracy of the parser in labeling law texts and TüBa-D/Z. As expected, the performance of the parser was lower if the test data were not derived from the same text type as the training data. The parser achieved labeling accuracies of 90.89% in TüBa-D/Z and 88.57% in law texts. Thus, the results proved that a statistical parser performed better on the test set that was derived from the same text type. This result confirms the findings of Versley (2005).

Test set	TüBa-D/Z	Swiss German-language law texts
Label accuracy	**90.89**% (211,376, 21,184)	88.57% (9,149, 1,181)

TABLE 2.14: Overall label accuracy of topological dependency relations of the statistical parser on Swiss German-language law texts and TüBa-D/Z:
label accuracy: % (# of correctly predicted tokens, # of incorrectly predicted tokens)

Since the domain difference in overall label accuracy is relatively small (2.32%), I further evaluate the parser with regard to each dependency topological label. Table 2.15 provides the precision, recall, and F1 scores of each topological dependency label, with the numbers of true positives, false negatives and false positives. To a certain extent,

[15] The data sets were the same as in the'Law400test' in Chapter 4 and 'TüBa20Eval' in Chapter 5. The 'Law400test' is a set of sentences that I annotated manually.

the differences in performance between the domains were expected. However, the result clearly shows that the topological fields and syntactic functions of arguments are analyzed particularly poorly in the law texts. This finding indicates that the parser lowered the performance in labeling wide-distance relations (i.e. topological fields) and middle-distance relations (i.e. syntactic functions of arguments) in law texts.[16] In contrast, dependency relations that were predicted better in law texts were related to the heads at a short distance: determiners (`DET`), the dependent of a preposition (`PN`) or a comparative conjunction (`KOM`), and coordinating elements (`KON` and `CJ`). The results of this comparison indicate that the domain difference between newspapers and law texts affected the accuracy of the error detection strongly, because the detection is based on syntactic categories of topological fields and arguments. Because we are primarily interested in achieving high precision in these syntactic categories for error detection, the impact of domain differences in parsing cannot be neglected.

[16] In this thesis, I categorize three types of syntactic relations based on the number of tokens between them: wide-distance, middle-distance and short-distance.

	TüBa20eval			Law400test		
	F_1	*(prec, rec)*	*(tp, fn, fp)*	F_1	*(prec, rec)*	*(tp, fn, fp)*
LB-V1	0.57	(0.51, 0.66)	(286, 278, 145)	**0.84**	(0.79, 0.89)	(33, 9, 4)
RB-V1	**0.57**	(0.50, 0.67)	(115, 115, 57)	0.55	(0.38, 1.00)	(9, 15, 0)
LB-V2	**0.94**	(0.96, 0.93)	(13206, 556, 1045)	0.93	(0.97, 0.90)	(426, 14, 47)
RB-V2	**0.88**	(0.91, 0.85)	(6155, 603, 1126)	0.83	(0.89, 0.78)	(239, 31, 67)
LB-VF-CONJ	**0.93**	(0.92, 0.93)	(2348, 206, 165)	0.91	(0.89, 0.93)	(78, 10, 6)
RB-VF-CONJ	**0.78**	(0.72, 0.85)	(2205, 842, 395)	0.64	(0.53, 0.81)	(71, 63, 17)
RB-VF-RC	**0.75**	(0.71, 0.80)	(1420, 575, 351)	0.68	(0.58, 0.80)	(77, 55, 19)
LXCOMP	**0.60**	(0.50, 0.77)	(313, 313, 96)	0.55	(1.00, 0.38)	(6, 0, 10)
RXCOMP	**0.85**	(0.90, 0.81)	(1077, 116, 250)	0.61	(0.85, 0.47)	(17, 3, 19)
TOP	**0.32**	(0.26, 0.39)	(9, 25, 14)	0.00	(0.00, 0.00)	(0, 10, 0)
VC	**0.96**	(0.96, 0.96)	(4441, 202, 185)	0.89	(0.86, 0.93)	(239, 40, 19)
SUBJ	**0.87**	(0.86, 0.88)	(14169, 2364, 2006)	0.84	(0.82, 0.85)	(496, 106, 85)
OBJA	**0.81**	(0.83, 0.78)	(7093, 1462, 1964)	0.76	(0.80, 0.73)	(245, 60, 91)
OBJD	**0.64**	(0.56, 0.75)	(797, 625, 259)	0.62	(0.50, 0.82)	(42, 42, 9)
OBJG	**0.23**	(0.16, 0.42)	(5, 26, 7)	0.00	(0.00, 0.00)	(0, 2, 0)
GMOD	0.87	(0.90, 0.84)	(4441, 480, 874)	**0.94**	(0.94, 0.93)	(364, 22, 26)
PN	**0.96**	(0.97, 0.95)	(19505, 652, 1025)	**0.96**	(0.94, 0.98)	(988, 58, 25)
PRED	**0.67**	(0.63, 0.71)	(1179, 691, 471)	0.28	(0.31, 0.25)	(4, 9, 12)
ZEIT	**0.66**	(0.61, 0.72)	(227, 147, 89)	0.00	(0.00, 0.00)	(0, 1, 3)
APP	**0.82**	(0.80, 0.84)	(6375, 1622, 1171)	0.00	(0.00, 0.00)	(0, 128, 17)
SUBJ=LB-VF-RC	**0.87**	(0.85, 0.89)	(1026, 176, 130)	0.82	(0.72, 0.94)	(64, 25, 4)
OBJA=LB-VF-RC	**0.66**	(0.69, 0.63)	(154, 70, 89)	0.53	(0.57, 0.50)	(4, 3, 4)
OBJD=LB-VF-RC	**0.74**	(0.68, 0.82)	(23, 11, 5)	0.33	(0.33, 0.33)	(1, 2, 2)
DET=LB-VF-RC	**0.65**	(0.53, 0.84)	(54, 48, 10)	0.60	(0.60, 0.60)	(6, 4, 4)
PN=LB-VF-RC	0.09	(0.05, 0.48)	(14, 268, 15)	**0.25**	(0.14, 1.00)	(1, 6, 0)
KON	0.89	(0.88, 0.90)	(6607, 941, 697)	**0.90**	(0.86, 0.94)	(592, 94, 41)
CJ	0.88	(0.88, 0.88)	(4258, 584, 607)	**0.89**	(0.90, 0.89)	(406, 47, 49)
DET	**0.98**	(0.99, 0.97)	(27317, 249, 875)	**0.98**	(0.97, 0.98)	(1562, 45, 26)
ATTR	**0.95**	(0.95, 0.96)	(13699, 789, 605)	0.93	(0.94, 0.91)	(658, 44, 62)
PP	0.91	(0.90, 0.91)	(16431, 1876, 1564)	**0.93**	(0.90, 0.96)	(877, 94, 36)
OBJP	**0.62**	(0.55, 0.73)	(1393, 1150, 522)	0.54	(0.54, 0.54)	(27, 23, 23)
KOM	0.86	(0.93, 0.80)	(1056, 74, 259)	**0.99**	(0.97, 1.00)	(34, 1, 0)
ADV	**0.94**	(0.94, 0.93)	(16065, 1002, 1162)	0.86	(0.83, 0.89)	(305, 63, 36)
-PUNCT-	**0.98**	(1.00, 0.96)	(33656, 75, 1521)	0.97	(0.97, 0.98)	(1077, 32, 27)

TABLE 2.15: F1 scores of topological dependency grammar labels (the statistical parser): prec = precision, rec = recall, tp = true positive, fn = false negative, fp = false positive

2.6 Conclusion

In this chapter, I presented a rule-based style error detection method for the automatic detection of violations of syntax-related rules in legislative style guidelines. I implemented an automatic error detection system using dependency grammar and topological dependency grammar parsers. By introducing the use of topological dependency representation in syntactic annotation, the accuracy of the error detection improved in the scores of F1, compared to a dependency grammar parser. The improvement was

thanks to the increase of recall, at a small cost in precision. The results indicate that the topological dependency representation is beneficial for the error detection task.

To gain insight into the performance of parsers in domain-specific texts, I tested both statistical parsers on the test data from the same corpus as the training data (TüBa-D/Z) and from Swiss German-language law texts. The results of the comparison showed that the differences between these two domains were considerable in the grammar categories at middle and wide distances, that is, syntactic functions of arguments and topological fields. Because error detection is mainly based on these grammar relations, the difference in domain strongly affected the performance of the error detection system.

In the following chapters (Chapters 3, 4, and 5), I will develop a domain-specific syntactic analyzer that is tailored to Swiss German-language law texts in order to improve the accuracy of the error detection system. In Chapter 6, I will then use the domain-specific syntactic analyzer for the error detection system presented in this chapter and test its accuracy.

Chapter 3

Approaches to the Automatic Annotation of Syntactic Structures

> *Handel mit gebrannten Wassern zu Trinkzwecken betreibt, wer solche verkauft, vermittelt oder auf andere Weise gegen Entgelt abgibt.*
>
> – Federal Law on Alcohol Art. 39 para. 1

In the next two chapters, I will develop an automatic syntactic analyzer for Swiss German-language law texts in order to improve the accuracy of the automatic style error detection presented in Chapter 2. In Chapter 2, I used state-of-the-art statistical parsers for the pre-processing syntactic annotation. Evaluation showed that the performance of the style error detection was strongly affected by that of the parsers, and the parser's performance was affected by the domain difference. In this chapter, I discuss the problem of domain adaptation in automatic syntactic annotation and propose supertagging as a solution.

3.1 Previous Approaches

3.1.1 Domain Adaptation Problems in Parsing

In any new domain, the initial lack of a large linguistically annotated corpus is a big challenge for parsing in NLP. It is well known that state-of-the-art statistical parsers learn the language patterns of the training data (mostly newspaper texts). However,

the learned models are usually not general enough to be applied to other domains and text types (Versley, 2005, Gildea, 2001, Sekine, 1997). Moreover, they are biased by the annotation schemata of the training data (Kübler et al., 2006).

Under these circumstances, the best approach is the manual annotation of a large amount of new domain data on which a parser can be trained. However, this is also the most cost-intensive solution.

The second approach is to apply semi-supervised domain adaptation methods, such as self-training (Goutam, 2012, Goutam and Ambati, 2012, McClosky and Charniak, 2008, Foster et al., 2007, Reichart and Rappoport, 2007, McClosky et al., 2006) and co-training (Goutam, 2012, Steedman et al., 2003, Sarkar, 2001). These approaches are used to improve the performance of statistical parsers without manually annotating in-domain data. In these approaches, in-domain training data is automatically generated by a statistical parser in self-training and by more than one parser in co-training. To ensure the quality of the automatically annotated training data, measurements such as ranking or confidence scores are used. These semi-supervised methods have the advantage that they can be used in other domains. However, the results of previous research on using these approaches are mixed. Both positive (McClosky and Charniak, 2008, Reichart and Rappoport, 2007, Foster et al., 2007, McClosky et al., 2006, Sarkar, 2001) and negative results (Attardi et al., 2013, Goutam and Ambati, 2012, Steedman et al., 2003) have been reported.

The third approach is active learning. In active learning, only a few effective training instances are deliberately selected so that the manual annotation of in-domain data is minimal. Two scenarios of active learning in parsing have been considered. First, a small amount of in-domain data was selected, annotated, and added to available, already annotated out-of-domain data. Attardi et al. (2013) achieved a significant improvement by adding 189 automatically parsed and manually corrected in-domain law sentences to 3,275 sentences collected from the Italian Syntactic-Semantic Treebank (ISST). The training instances were selected by using the likelihood measurement returned by a parser. In this approach, the instances that were parsed with less certainty were assumed the optimal in-domain training instances. In the second scenario, in-domain-like instances were selected from already annotated out-of-domain data and then added to in-domain training data. Khan et al. (2013b) demonstrated that adding existing out-of-domain data similar to the in-domain data improved the accuracy of a parser in analyzing in-domain data. They exploited perplexity in a 5-gram language model to approximate similarity. Low perplexity indicated the similarity of the out-of-domain sentences to the in-domain data.

The fourth approach has been used to address issues of domain-specific unknown words in the preprocessing of parsing. For example, to cope with noisy social media data, input texts were normalized into regular word forms by correcting misspellings before parsing (Khan et al., 2013a, Foster, 2010). In the biomedical domain, suffix guessing (Pyysalo et al., 2006, Aubin et al., 2005), the extension of lexicons (Pyysalo et al., 2006, Aubin et al., 2005) or the use of a domain-specific POS tagger (GENIA tagger) (Pyysalo et al., 2006) were reported.

The first three approaches are intended to improve statistical parsers by configuring training data. The first approach is not realistic in many cases. The second approach is adaptable to any domains and does not require any manual annotation. However, the results are discouraging. The third approach is the best approach to use in a realistic scenario for domain adaptation if the in-domain resources are sparse. The fourth approach is also a reasonable approach in this respect. However, this approach is not directly applicable to the legislative domain. Law texts are professionally written and edited (in contrast to e.g. user-generated content of social media). The target audience of law texts consists of lay persons (unlike e.g. bio-medical texts). To estimate the unknown word issue in the domain, I computed the proportion of words and word types in Swiss German-language law texts that are not found in TüBa.[1] 13.65% of the words (647,281 out of 4,742,878) and 80.32% of the word types (98,676 out of 122,857) were unknown words and unknown word types for the domain. The words are unknown, mostly because of compounds such as *Sozialversicherungsgericht* or *Bodengasmessungen* and because of the regional spelling differences, that is, for example, the double s problem, e.g. *Beschleunigungsmassnahmen* vs. *Beschleunigungsmaßnahmen*. To estimate the unknown word issue in an NLP pipeline, I estimated how many words are returned as unknown by an off-the-shelf part-of-speech tagger (here: TreeTagger (Schmid, 1999)) and a morphological analyzer (here: Gertwol (Haapalainen and Majorin, 1994)). 5.7% of words and 48.89% of word types were unknown by the TreeTagger, and 12.72% of words and 10.65% of word types by Gertwol. These results indicate that unknown words are not the main issue in the domain, as long as words are analyzed morphologically.[2]

[1] In this estimation, I used all tokens in Swiss German-language law texts and TüBa.

[2] As domain-specific pre-processing, it is required to convert enumeration structures into newspaper-like sentences. However, enumeration structures in law texts are often complex. The following sentence is a typical example. Enumeration structures might be segmented based on colons, semicolons, or periods, which are noisy in the input law texts (e.g. b1, b2, and b3 are separated by commas instead of semicolons).

(i) In diesem Gesetz bedeuten:

a. Würde: Eigenwert des Tieres, der im Umgang mit ihm geachtet werden muss. Die Würde des Tieres wird missachtet, wenn eine Belastung des Tieres nicht durch überwiegende Interessen gerechtfertigt werden kann. Eine Belastung liegt vor, wenn dem Tier insbesondere

In the domain, the main issue is syntactic complexity. Law texts tend to contain complex syntactic structures due to the complexity of legal content. Typically, law texts describe legal conditions and consequences. Conditions are syntactically realized as verb-first clauses, subordinating clauses with discourse connectors such as *wenn, falls* or *solange*, or relative clauses (cf. Section 2.2.2). As a result, sentences typically contain more than one clause. The sentence (1) exhibits a typical example in the domain. It comprises of a verb-first conditional clause, a main clause, and a subordinating clause. This type of syntactic complexity causes parsing problems because of wide-distance syntactic relations. The sentence (1) is parsed as (2) by the statistical graph-based Bohnet parser I used in the previous chapter. The correct tree is illustrated as (3). The dependency tree (2) returned from the parser contains errors with regard to dependency relations at wide distances, that is, those of verbs. Parsers trained on a newspaper corpus cannot cope with the syntactic analysis of the word such as *bestimmt* in the sentence (1). Besides the difficulties of wide-distance mapping between discontinuous verbs, the problem further lies in the morphosyntactic ambiguity of verbs. In this example (1), *bestimmt* can be analyzed morphosyntactically as adverb or verb. The error is rooted in that it is used contextually differently in the domain: Typically in newspapers, the word *so* is used as adverb, modifying and intensifying adjective, e.g. *so schöne Katzen*, or adverb, e.g. *so bestimmt nicht*. Thus, the word after the intensifier *so* is more likely to be adjective or adverb. In law texts, *so* is, however, more likely to be used as resultative to refer to verb-first conditional clauses (cf. the sentence (1)). In this case, the word afterwards (here: *bestimmt*) is the main verb. Thus, the unknown *use* of words and the morphosyntactic ambiguity are the underlying problems to parse law texts.

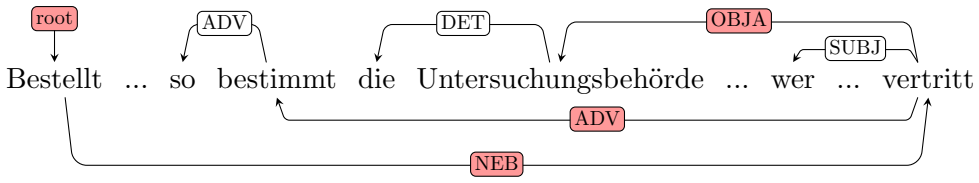
Schmerzen, Leiden oder Schäden zugefügt werden, es in Angst versetzt oder erniedrigt wird, wenn tief greifend in sein Erscheinungsbild oder seine Fähigkeiten eingegriffen oder es übermässig instrumentalisiert wird;

b. Wohlergehen: Wohlergehen der Tiere ist namentlich gegeben, wenn:
   1. die Haltung und Ernährung so sind, dass ihre Körperfunktionen und ihr Verhalten nicht gestört sind und sie in ihrer Anpassungsfähigkeit nicht überfordert sind,
   2. das artgemässe Verhalten innerhalb der biologischen Anpassungsfähigkeit gewährleistet ist,
   3. sie klinisch gesund sind,
   4. Schmerzen, Leiden, Schäden und Angst vermieden werden;

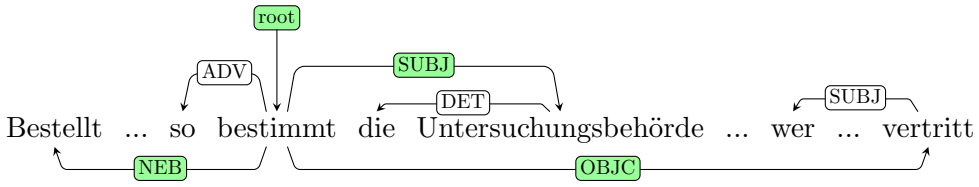
c. Tierversuch: jede Massnahme, bei der lebende Tiere verwendet werden mit dem Ziel:
   1. eine wissenschaftliche Annahme zu prüfen,
   2. die Wirkung einer bestimmten Massnahme am Tier festzustellen,
   3. einen Stoff zu prüfen,
   4. Zellen, Organe oder Körperflüssigkeiten zu gewinnen oder zu prüfen, ausser wenn dies im Rahmen der landwirtschaftlichen Produktion, der diagnostischen oder kurativen Tätigkeit am Tier oder für den Nachweis des Gesundheitsstatus von Tierpopulationen erfolgt,
   5. artfremde Organismen zu erhalten oder zu vermehren,
   6. der Lehre sowie der Aus- und Weiterbildung zu dienen.

(1) Bestellt das Unternehmen nicht innert angemessener Frist einen derartigen Vertreter, so bestimmt die Untersuchungsbehörde oder das Gericht, wer von den zur zivilrechtlichen Vertretung befugten Personen das Unternehmen im Strafverfahren vertritt.

(2) Dependency parser output:



(3) Correct parse tree:



3.1.2 Partial Parsing as Application-Oriented Approach

In this thesis, another approach to the domain adaptation problem is considered viable: the use of partial (or shallow) parsing, i.e. chunking, part-of-speech (POS) tagging, and supertagging. In this approach, the syntactic representations to be analyzed are simpler than those of full parsing.

Chunking (Abney, 1996) is the segmentation of basic non-recursive phrases, such as NP, VP, PP, S, and so on.[3] Two types of chunking in German have received particular attention in the literature: noun chunking[4] (Mahlow and Piotrowski, 2010, Schiehlen, 2002, Kermes and Evert, 2002, Schiehlen, 2002, Schmidt and Schulte im Walde, 2000,

[3] In the CoNLL Shared Task 2000 (Sang and Buchholz, 2000), chunks are defined as non-recursive and non-overlapping segments. Each chunk contains one head word and each word belongs to only one chunk.

[4] In noun chunking, two types of noun chunks have been proposed: non-recursive NP (Mahlow and Piotrowski, 2010, Schiehlen, 2002) and recursive NP (Kermes and Evert, 2002, Schiehlen, 2002, Schmidt and Schulte im Walde, 2000, Skut and Brants, 1998a,b). In German, reduced relative clauses are prenominal and embedded in noun phrases, which poses problems for the definition of NP in noun chunking (cf. Kübler et al., 2010, Hinrichs, 2005)

Skut and Brants, 1998b,a) and clause chunking (Cheung and Penn, 2009, Braun, 2003, Frank et al., 2003, Becker and Frank, 2002, Klatt, 2002, Müller and Ule, 2002, Veenstra et al., 2002, Neumann et al., 2000). The recognition of topological fields in German is considered quasi clause chunking.

POS tagging assigns a part of speech (i.e. coarse morphosyntactic category) to each word. In the research on POS tagging, a POS tagger developed by Voutilainen (1995b) in Constraint Grammar is the benchmark English tagger (accuracy above 99%), whereas the accuracy of data-driven machine learning methods has remained above 97% (Manning, 2011). In German, the widely used statistical POS tagger, TreeTagger (Schmid, 1999), with an accuracy of 92.37%, is slightly better than the rule-based tagger (i.e. transformation-based tagger developed by Brill (1992)) with an accuracy of 91.65% (Volk and Schneider, 1998). More recently, an accuracy of 97.5% was reported (Schmid, 1999).[5]

Supertagging is an "almost parsing" approach (Bangalore and Joshi, 1999), in which the supertags represent rich linguistic information, such as valence, voice, and grammatical functions (Foth et al., 2010, Harper and Wang, 2010). A parser then needs "only [to] combine the individual supertags" (Bangalore and Joshi, 1999). Supertagging approaches are employed in preprocessing for various parsing grammar formalisms, such as dependency grammar (Foth et al., 2010, Harper and Wang, 2010), lexicalized tree-adjoining grammar (LTAG)(Bangalore and Joshi, 1999), head-driven phrase structure grammar (HPSG)(Zhang et al., 2009) and combinatory categorial grammar (CCG)(Clark, 2011, Curran et al., 2006) or semantic parsing (Chen, 2010). Supertagging approaches are also used in various applications, such as information retrieval, information extraction, text simplification (see details in Bangalore and Joshi (1999)) and dialogue systems (Harbusch et al., 2010). Foth et al. (2010) developed a supertagger for the dependency grammar parsing of German texts. In this supertagger, dependency grammar labels and a part of dependency paths are recognized by using a TnT (Brants, 2000).

[5]However, it was also reported that the accuracy of TreeTagger dropped below 90.7% in a real-life scenario of tagging web data (Giesbrecht and Evert, 2009).

3.2 My Approach: Topological Dependency Supertagging

Compared to full parsing, a partial parsing approach has the advantage of high accuracy (cf. Abney, 1997).[6] The accuracy of annotation and the complexity of the syntactic representations to be analyzed are often a trade-off. In my case, in which syntactic annotation is used directly in an application, accuracy is more relevant than completeness is. With the reduction in the complexity of syntactic representations, the effort required in manual annotation is decreased, which is an additional advantage of the partial parsing approach.

In my work, I follow a supertagging approach. In this approach, essential syntactic structures and functions are automatically annotated by using tagging methods. The linguistic representation that I used in the application is the topological dependency grammar that was presented in Chapter 2, which showed that topological dependency grammar was beneficial in style error detection.

In the following sections, I provide an overview of the representation of topological dependency tagging and then present a brief overview of supertagging methods. Finally, I present the architecture of the supertagger used to parse the German language.

3.2.1 Topological Dependency Grammar

The topological dependency tag set is presented in Chapter 2 (cf. Table 2.9 and Table 2.10). The supertag set comprises two types of grammars: dependency grammar (Foth, 2005) and topological fields (Höhle, 1986). The supertagger assigns a dependency grammar tag *or* or a topological field tag to each token. The tags capture basic German linear clause structures and types, as well as syntactic functions.

The target of supertagging is oriented to a CoNLL format of dependency grammar parsing (cf. Table 3.1).[7] In the CoNLL format, dependency grammar relations are assigned to dependents:[8] In full parsing, dependency relations (`DEPREL`) to the heads

[6] Partial parsing is a "cover term for a range of different techniques for recovering some but not all of the information contained in a traditional syntactic analysis. Partial parsing techniques [...] aim for reliability and robustness in the face of the vagaries of natural text, by sacrificing completeness of analysis and accepting a low but non-zero error rate (Abney, 1997)"

[7] The CoNLL format is a representation defined in CoNLL-X Shared Task 2006, Multi-lingual Dependency Parsing (`http://ilk.uvt.nl/conll/index.html` (last visited: 27/03/2015))

[8] `CPOSTTAG` is a coarse POS tag; `FEAT` is a morphological and syntactic feature; `HEAD` is the ID of the head of current token; and `DEPREL` is the dependency relation to the `HEAD`. Two fields, `PHEAD` and `PDEPREL`, are not included. `PHEAD` is the projective head of the current token, and `PDEPREL` is the dependency grammar relation to `PHEAD`.

are identified with the ID of the heads (`HEAD`). The supertagging addressed in this thesis is focused on the assignment of `DEPREL`, which corresponds to the simplest supertag set in Foth et al. (2010). `HEAD` is ignored because this task belongs to full parsing.

ID	WORD	LEMMA	CPOSTAG	POSTAG	FEAT	HEAD	DEPREL	-	-
1	Dieses	diese	ART	PDAT	_	2	DET	_	_
2	Gesetz	Gesetz	N	NN	_	3	SUBJ	_	_
3	untersteht	unterstehen	V	VVFIN	_	0	ROOT	_	_
4	dem	die	ART	ART	_	6	DET	_	_
5	fakultativen	fakultativ	ADJA	ADJA	_	6	ATTR	_	_
6	Referendum	Referendum	N	NN	_	3	OBJD	_	_
7	.	.	$.	$.	_	6	-PUNCT-	_	_

TABLE 3.1: Dependency grammar representation in CoNLL

3.2.2 Methods

Supertagging is based on tagging methods. They are widely applied in NLP, such as in POS tagging (e.g. Toutanova and Manning, 2000), parsing (Finkel et al., 2008, Carreras et al., 2006), named entity recognition (Faruqui and Padó, 2010, Finkel et al., 2005, Miller et al., 2004), and morpheme/word segmentation (Zhao et al., 2006). The concept of tagging is that the linguistic category of a token is dependent not only on the token but also on its linear context. Commonly, two main methods have been considered: rule-based taggers and statistical taggers. In rule-based tagging (Karlsson et al., 1995, Brill, 1992, Klein and Simmons, 1963, Harris, 1962), rules are if-then conditions that are applied categorically to select a plausible category (e.g. POS tag) among all possible categories. Possible categories are generated based on a dictionary, suffix-guesser, morphological analyzer, or an annotated corpus. Examples of such rules are 'assign category x if previous token is word y' or 'assign category x if the token matches with permissive category sequences between article and verb (i.e. adjective-noun, noun-noun and noun-adverb)'. Most algorithms (Brill, 1992, Klein and Simmons, 1963, Harris, 1962) are based on local context (1 to 3 tokens left and right from current token), whereas constraint grammar (Karlsson et al., 1995) allows an unlimited global context. In supervised statistical tagging, ambiguity is resolved in a probabilistic manner by using an annotated corpus. The statistical models are based on the properties of current tokens (e.g. word form, suffix) and those of the local contexts. These two methods have the following advantages and disadvantages:

- **Performance**: Rule-based taggers and supervised statistical taggers are competitive with regard to their performance. For example, a rule-based English POS tagger developed by Voutilainen (1995b) is the benchmark English tagger

(the accuracy of above 99%), whereas supervised statistical taggers have remained at the accuracy of approximately 97% (Manning, 2011). The accuracy of unsupervised statistical taggers is lower than the accuracy of supervised taggers (cf. Christodoulopoulos et al., 2010, Merialdo, 1994)

- **Linguistic principle and knowledge**: Linguistic principles are described as grammar rules (if-then conditions) in tagging methods. In a rule-based approach, grammar rules are applied in a categorical manner. Therefore, they enable the implementation of the intended linguistic principles. However, the coverage of grammar rules is dependent on the linguistic knowledge of the grammar writers. Moreover, the complexity of grammar rules is limited if the rules are hand-written.[9] In a feature-rich statistical tagger, linguistic knowledge is described as features. The features are based on a simple template of if-then conditions. Because the features are automatically extracted for the template, the complexity of the features is not an issue (except the speed of training and testing and the memory requirement). Moreover, a feature-rich statistical tagger can integrate linguistic principles in a probabilistic manner, which is advantageous in modeling probabilistic and usage-based grammars. However, the learned probabilistic models can conflict with the intended linguistic constraints. In addition, the coverage of features is dependent on the training data.

- **Resource**: A rule-based tagger often does not directly induce patterns from a manually annotated corpus and thus does not need any annotated corpora, or at least, it does not need a large amount of annotated corpora for the development of a tagger.[10] Supervised statistical taggers require an annotated corpus, whereas unsupervised statistical taggers do not need any annotated data.

Considering these advantages and disadvantages, in the next section, I propose a hybrid architecture of the supertagger to overcome the shortcomings of these two approaches.

3.2.3 Architecture

German is a relatively morphologically rich language with a relatively free order. Morphology contributes partly to the encoding of grammatical functions (i.e. syntactic functions of arguments). Morphological markings are too ambiguous to be linked directly

[9] In the Brill tagger, the rules are in a form of templates and learned from annotated corpora. In this respect, the Brill tagger is not a typical rule-based tagger.

[10] The Brill tagger is an exception and learns patterns from an annotated corpus.

to syntactic functions; morphological syncretism is pervasive in German. At the same time, German is a relatively free word order language in which word order encodes grammatical functions only to a certain degree. For example, the sentence *Maria liebt Peter* can be reformulated as *Peter liebt Maria*, meaning Maria loves Peter, if only the prosody is marked. Thus, in German, morphology and syntax combine to encode grammatical functions. This type of language is regarded as a non-configurational language in linguistics (Hale, 1982) or a morphologically rich language in parsing (Tsarfaty et al., 2010). In contrast, English is regarded as a configurational language in which the word order encodes grammatical functions.

For the development of a German parser, the peculiarities of the language raise the question of how to build a system that can model the interaction between morphology and syntax. In statistical parsing, the integration of morphosyntactic case information into statistical parsing models is a main issue in the architecture of parsing (e.g. Seeker and Kuhn, 2011, Versley and Rehbein, 2009). In probabilistic context free grammar (PCFG) parsing approaches, it has been proposed to integrate morphosyntactic information by treebank transformations (Versley and Rehbein, 2009, Dubey, 2005, Schiehlen, 2004). Recently, Seeker and Kuhn (2013) developed a dependency-grammar parser in a joint architecture of case morphology and syntax. In the joint parsing approach, morphological and syntactic information are shared and jointly inferred (Lee et al., 2011, Goldberg and Tsarfaty, 2008, Cohen and Smith, 2007, Tsarfaty, 2006). This approach differs from the traditional pipeline architecture in that morphological analysis (e.g. morpheme segmentation and morphosyntactic tagging in Hebrew) and syntactic parsing are processed simultaneously, and the former is not deterministically predicted before the processing of the latter. The parser developed by Seeker and Kuhn (2013) uses the morphosyntactic case as a filter instead of completely disambiguated and predicted case features, and it integrates morphological hard rules into statistical parsing.

In another approach, parsers of German were built according to a pipeline architecture in which shallow parsing and deep parsing were sequentially processed (Sennrich et al., 2013, 2009, Foth and Menzel, 2006, Müller, 2005, Trushkina, 2004, Daum et al., 2003, Frank et al., 2003, Hinrichs et al., 2002). In these approaches, morphological ambiguity is incrementally solved in linguistically motivated components. Rule-based and statistical-based methods are often combined to solve ambiguity (Sennrich et al., 2013, Foth and Menzel, 2006, Trushkina, 2004, Daum et al., 2003, Frank et al., 2003, Crysmann et al., 2002, Hinrichs et al., 2002). The topological field model is often integrated in deep parsing (Müller, 2005, Trushkina, 2004, Crysmann et al., 2002, Frank et al., 2003, Hinrichs et al., 2002). In Foth and Menzel (2006), a supertagger is integrated into a hybrid dependency parser.

In this thesis, I will develop a topological dependency supertagger that models the interaction between morphology and syntax in a rule-based and statistical method. Following previous work, I consider that the underlying problems in syntactic annotation in German are rooted in morphosyntactic ambiguity. Problems with the syntactic annotation of domain-specific texts, such as law texts, begin with morphosyntactic ambiguity. To handle morphosyntactic ambiguity, I pursue a rule-based method. A rule-based tagger does not need any manually annotated data for development, and it has proved to be accurate in the field of POS tagging. In my work, a rule-based tagger is introduced for the preprocessing of the statistical tagger, thereby eliminating implausible morphosyntactic analyses and assigning possible syntactic functions. In the rule-based tagger, inviolable hard grammar constraints are integrated into the model. Because the hard constraints do not encode all syntactic functions, a statistical tagger is sequentially introduced to assign a supertag to each token among the morphosyntactically possible analyses, filtered by the rule-based tagger. In the statistical tagger, violable probabilistic soft grammar constraints are modeled. In other words, the issues of free word order in German are handled by a statistical method. In this respect, my work is similar to Trushkina (2004). In Trushkina (2004), the morphological agreement was handled by a rule-based tagger, and morphologically unsolvable ambiguity (i.e. word order) was solved statistically. In her work, the issue of free word order was solved by transforming the annotation of trees to project morphosyntactic information in a PCFG parser. Unlike Trushkina (2004), in Müller (2005), morphologically unsolvable ambiguity is resolved by using a default word order in a rule-based finite-state parsing approach. In addition, this thesis explores the configuration of training data in the statistical tagger to overcome the lack of in-domain resources. In this respect, my work is similar to Attardi et al. (2013).

3.3 Conclusion

In this chapter, I motivated the development of a supertagger for Swiss German-language law texts in order to boost the performance of the style error detection system that was developed in Chapter 2. To overcome the lack of in-domain resources for the development of a highly accurate syntactic analyzer that can be used in a real application, I proposed a topological approach to dependency grammar supertagging. Furthermore, I presented an architecture of the hybrid supertagger, which combines a rule-based component and a statistical component to resolve morphosyntactic ambiguity and issues of free word order in German.

In the next two chapters (Chapter 4 and 5), I will present the hybrid supertagger. In Chapter 4, I introduce the rule-based tagger. In Chapter 5, I present the statistical tagger which is then combined with the rule-based tagger described in Chapter 4.

Chapter 4

Rule-Based Supertagger

> *Die Vorsorgeeinrichtung muss einen dem Geld-, Kapital- und Immobilienmarkt entsprechenden Ertrag anstreben.*
>
> – Ordinance on Old-Age, Survivors' and Disability Insurance Art. 51

This chapter describes the development of a method to automatically analyze topological dependency grammar relations (cf. Chapter 2 and 3).[1] In the development of this method, the fundamental challenges lie in the morphosyntactic and syntactic ambiguity of German. In this chapter, I provide an analysis of the problem of ambiguity in detail (Section 4.1) and present a rule-based supertagger that reduces morphosyntactic ambiguity and maps topological dependency tags (Section 4.2).

4.1 Supertagging Problem

4.1.1 Morphosyntactic Ambiguity

Morphosyntactic ambiguity is pervasive in German. On average, in Swiss German-language law texts, one word has 6.0 morphological interpretations (6.5 per word type) according to Gertwol, the German morphological analyzer (Haapalainen and Majorin, 1994). Gertwol performs morphological analyses and delivers a bundle of coarse and fine-grained morphosyntactic features (cf. Gertwol's tag set in Appendix A). For example, the word *einen* has nine morphosyntactic interpretations (or readings):

[1] This chapter is based on Sugisaki and Höfler (2013a,b)

(1) *einen*

"man"	PRON PERS SG3 AKK
"ein"	ART INDEF SG AKK MASK
"einer"	PRON INDEF SG AKK MASK
"ein∼en"	V INF
"ein∼en"	V IND PRÄS PL1
"ein∼en"	V IND PRÄS PL3
"ein∼en"	V KONJ PRÄS PL1
"ein∼en"	V KONJ PRÄS PL3
"ein"	PL DAT STARK

Table 4.1 illustrates the frequency distribution of morphosyntactic interpretations in Swiss German-language law texts.[2] Only 20% of tokens (18% of word types) are unambiguous, that is, they have only one morphosyntactic reading. Moreover, 18% of words (18% of word types) have more than 10 morphosyntactic readings.

# of morphological readings	**Proportion**	**# of tokens**	**Proportion**	**# of word types**
1	20%	4467	18%	722
2	10%	2,164	7%	276
3	8%	1,662	9%	337
4	18%	4,010	21%	803
5	5%	1,083	7%	289
6	6%	1,281	4%	171
7	2%	368	3%	114
8	7%	1,452	6%	245
9	3%	553	5%	211
10	3%	583	2%	87
11 ∼ 155	18%	4,355	18%	652
Total	100%	21,978	100%	3,907

Table 4.1: Frequency distribution of fine-grained morphosyntactic readings in Swiss German-language law texts

To investigate the frequency distribution, I divided morphosyntactic features into eight coarse morphosyntactic categories. They are grouped into lexical categories and functional categories:

[2]The tokens for which Gertwol fails to provide any analyses are excluded. Gertwol returns 155 morphological readings for the word *Pfandrechtseinträgen*.

Lexical categories	
V	Verb, including verbal prefix
N	Full noun, including truncated noun
ADJ	Attributive and predicative adjective, including number
ADV	Adverb, including pronominal adverb
Functional categories	
PREP	Preposition and postposition
CONJ	Subordinating, infinitive and coordinating conjunction
PRON	Personal, possessive, indefinite, interrogative, demonstrative and relative pronoun
DET	Article and demonstrative, possessive and interrogative determiner

TABLE 4.2: Coarse morphosyntactic categories

Table 4.3 shows the frequency distribution of the coarse morphosyntactic categories generated by Gertwol. The majority of tokens are not ambiguous at this level: 61.65% of words (74.96% of word types) have one coarse morphosyntactic interpretation. The most ambiguous words are ambiguous across five coarse categories. For example, the word *einer* is analyzed as a determiner, verb, number (here: adjective), noun, and adverb. The word *nahe* is analyzed as an adjective, noun, adverb, preposition, and verb.

# of coarse morphosyntactic categories	**Proportion**	**# of tokens**	**Proportion**	**# of word types**
1 category	61.65%	13,549	74.96%	2928
2 categories	30.44%	6,690	23.30%	910
3 categories	6.37%	1,401	1.54%	60
4 categories	1.49%	327	0.13%	5
5 categories	0.05%	10	0.08%	3
Total	100%	21,977	100%	3,907

TABLE 4.3: Frequency distribution of coarse morphosyntactic categories in Swiss German-language law texts

Table 4.4 illustrates the ambiguity rate of paired coarse morphosyntactic categories. Almost all the pairs of coarse morphosyntactic categories are ambiguous. In particular, lexical categories, such as verbs, nouns, adjectives, and adverbs, are highly ambiguous. In word type, 95.6% of all binary ambiguities (49.13% in tokens) entail ambiguities between verbs, nouns, and adjectives. Moreover, 98.23% of all paired ambiguities in word type (89.85% in token) involve five morphosyntactic categories and the following pairs:

(2) a. verb vs. noun (13.10% in token, 19.66% in word type)
b. verb vs. adjective (11.24% in token and 21.92% in word type)

c. noun vs. adjective (24.79% in token and 54.02% in word type)

d. pronoun vs. determiner (40.72% in token and 2.63% in word type)

	N	ADJ	ADV	PREP	CONJ	PRON	DET
V	**13.10** (**19.66**)	**11.24** (**21.92**)	-	0.88 (0.14)	0.13 (0.03)	0.70 (0.09)	0.79 (0.1)
N		**24.79** (**54.02**)	0.12 (0.23)	0.13 (0.12)	-	3.10 (0.36)	2.06 (0.11)
ADJ			0.12 (0.19)	0.25 (0.07)	0.03 (0.02)	0.14 (0.02)	0.18 (0.03)
ADV				0.97 (0.13)	0.19 (0.04)	0.18 (0.04)	-
PREP					0.17 (0.06)	-	-
CONJ						-	-
PRON							**40.72** (**2.63**)

TABLE 4.4: Percent proportion of binary coarse morphosyntactic ambiguity in Swiss German-language law texts: token (word types)

Table 4.5 provides the average number of morphosyntactic readings generated by Gertwol for each coarse morphological category, per token and per word type. The table shows the ambiguities in coarse morphological categories at the level of fine-grained morphosyntactic interpretations (e.g. person, number, gender, case, mood, and tense). The table also shows that verbs, nouns, adjectives, pronouns, and determiners are ambiguous at the level of fine-grained morphosyntactic features.

Coarse morphological category	**Averaged # of fine-grained morphological readings (words)**	**Averaged # of fine-grained morphological readings (word types)**
Verb	**3.2**	**4.3**
Noun	**5.1**	**5.0**
Adjective	**5.5**	**6.4**
Adverb	1.0	1.0
Preposition	1.5	1.5
Conjunction	1.1	1.2
Pronoun	**5.9**	**4.1**
Determiner	**3.5**	**3.1**

TABLE 4.5: The average number of fine-grained morphosyntactic readings returned by Gertwol

In summary, morphosyntactic ambiguity is widespread in German. In particular, the majority of words (61.65%) are not ambiguous at the level of coarse morphosyntactic

categories in law texts; they have one possible coarse morphosyntactic category. The remaining 30.35% of words are mostly ambiguous among lexical categories and among functional categories. At the level of a fine-grained morphosyntactic analysis, six morphosyntactic readings per word are found on average. Thus, the major problems of ambiguity in German are (i) the coarse ambiguity among functional categories and (ii) the one among lexical categories and (iii) the fine-grained ambiguity. If the reader is interested in the qualitative analysis of morphosyntactic ambiguity, I suggest consulting Appendix F. In the next section, I analyze the syntactic ambiguity between morphosyntactic categories and topological dependency categories. Based on the analysis, I will define the required morphosyntactic granularity to develop a topological dependency grammar tagger.

4.1.2 Syntactic Ambiguity

Morphosyntactic and syntactic categories do not have a one-to-one relationship. Table 4.6 illustrates the rates of ambiguity in mapping from Gertwol's coarse morphosyntactic categories to syntactic tags. The table indicates that all mappings are ambiguous. They are in a one-to-many or many-to-many relation because all morphosyntactic categories can be mapped to the coordinated elements (`KON` or `CJ`). For example, determiner or subordinating conjunction can be coordinated, such as *Der and die* in the sentence (3-a) or *ob, wann und wie* in the sentence (3-b). Excluding `KON` and `CJ`, the majority of coarse morphosyntactic categories are not ambiguous and can be mapped directly to syntactic categories.[3] Among the coarse morphosyntactic categories, ambiguous categories are nouns, verbs, and prepositions. The mapping rates are 1:13, 1:9, 1:5, respectively. Thus, nouns can be mapped to one of 13 possible syntactic categories, whereas verbs and prepositions can be linked to one of 9 and 5 syntactic tags, respectively.

(3) a. **Der oder die** Verantwortliche für Logistik und Sicherheit ist für die sichere Aufbewahrung der Akten zuständig und betreut die dafür vorgesehenen Räumlichkeiten.[4]

b. Der Urheber oder die Urheberin hat das ausschliessliche Recht zu bestimmen, **ob, wann und wie** das Werk verwendet wird.

[3] In Gertwol, adjectives are analyzed as `A` and `A(PART)`. To differentiate predicative and attributive, the number features are used. The number features are absent in predicative adjectives, whereas they are present in attributive adjectives. Conjunctions are differentiated in subordinating, comparative, infinitive, and coordinating conjunctions; Gertwol provides `uKONJ`, `sKONJ`, `iKONJ`, `nKONJ`, respectively.

[4] All examples in this chapter are derived from Swiss German-language law texts (cf. Section 1.5) unless they are marked otherwise. I do not translate or gloss the example sentences in German because it would not help to understand the language-specific problems discussed here.

Coarse Morphosyntactic categories (Gertwol tag)	**Syntactic tags**	**Mapping Rate**
Noun (`S, S(A), S(PART)`)	SUBJ, OBJD, OBJA, OBJG, GMOD, EXPL, APP, PRED, ZEIT, GRAD, PN, KON or CJ	1:13
Verb/predicate (`V, A, A(PART)`)	LB-V1, LB-V2, RB-V1, RB-V2, RB-VF-CONJ, RB-VF-RC, VC, TOP or RXCOMP	1:9
Adposition (`PRÄP` and `PRONADV`)	OBJP, PP, PRED , KON or CJ	1:5
Verbal prefix (`PRÄF`)	RB-V2, KON, CJ	1:3
Determiner (`ART` and `DET`)	DET, KON or CJ	1:3
Adjective (`A` and `A(PART)`)	ATTR , KON or CJ	1:3
Adverb (`ADV` and `PRONADV`)	ADV , KON or CJ	1:3
Subordinating conjunction (`uKONJ`)	RB-VF-CONJ, KON, CJ	1:3
Comparative conjunction (`sKONJ`)	KOM, KON or CJ	1:3
Infinitive conjunction (`iKONJ`)	LXCOMP, KON or CJ	1:3
Coordinating conjunction (`nKONJ`)	CJ	1:1

TABLE 4.6: Mapping of syntactic categories from coarse morphosyntactic categories

The ambiguities of linking from morphosyntactic categories to syntactic categories are reduced by the distinctions of the fine-grained morphosyntactic categories of nouns and verbs: case and finiteness. Table 4.7 illustrates that specifying the case features of nouns reduced the ambiguity rate of 1:13 to a range between 1:5 and 1:7. The case features of nouns provide cues for the distinction between arguments, such as subjects (`SUBJ`), but they do not differentiate adjuncts such as coordinating elements (`KON` and `CJ`) and appositions (`APP`). The distinction between finite (indicative, conjunctive, and imperative) and nonfinite verbs (participle, infinitive, and zu-infinitive) reduced the ambiguity rate from 1:9 to a range between 1:4 and 1:5.

Fine-grained morphosyntactic categories (Gertwol tag)	**Syntactic tags**	**Mapping Rate**
Nominative noun (`S NOM`)	SUBJ, EXPL, PRED, APP, PN, KON or CJ	1:7
Accusative noun (`S AKK`)	OBJA, ZEIT, GRAD, APP, PN , KON or CJ	1:7
Genitive noun (`S GEN`)	OBJG, GMOD, APP, PN , KON or CJ	1:6
Dative noun (`S DAT`)	OBJD, APP, PN, KON or CJ	1:5
Nonfinite verb (`V INF/PART/INF zu`)	TOP, VC, RB-V1, RB-V2, RXCOMP	1:5
Finite verb (`V IND/IMP/KONJ`)	LB-V2, LB-V1, RB-VF-CONJ, RB-VF-RC	1:4

TABLE 4.7: Mapping of syntactic categories from fine-grained morphosyntactic categories

In summary, the majority of coarse morphosyntactic categories are syntactically unambiguous and can be directly mapped to topological dependency grammar labels. Highly ambiguous coarse morphosyntactic categories are verbs, nouns, and prepositions. The resolution of finiteness and case features reduced their syntactic mapping ambiguity.

In the next section, I will develop a supertagger that reduces morphosyntactic ambiguity and performs syntactic mapping. For the development, the following six major morphosyntactic ambiguities (4) need to be resolved:

(4)
- ***The coarsest morphosyntactic ambiguity***:
  (i) Lexical or functional category? (e.g. *sein, einen, Statt*)
- ***Coarse morphosyntactic ambiguity in functional categories***:
  (i) Determiner, demonstrative pronoun or relative pronoun? (e.g. *der, die, das*)
  (ii) Subordinating conjunction, preposition, verbal prefix or infinitive conjunction? (e.g. *während, bis, mit, ohne*)
- ***Coarse morphosyntactic ambiguity in lexical categories***:
  (i) Verb, noun, adjective or adverb? (e.g. *Verfahren, bestimmt*)
- ***Fine-grained morphosyntactic ambiguity***
  (i) Finite or nonfinite verb? (e.g. *studiert, schreiben*)
  (ii) Nominative, genitive, accusative or dative case? (e.g. *Katze, Katzen*)

4.2 Constraint-Grammar-Based Topological Dependency Supertagger

In this section, I develop a topological dependency supertagger based on morphosyntactic disambiguation. The supertagger is geared to Swiss German-language law texts. I use Constraint Grammar (CG) (Karlsson et al., 1995) as a framework. A CG-based tagger is rule-based and comprises a set of hand-written rules. In terms of the task performed here, CG-tagging has the following advantages and limitations:

(5)
- A rule-based tagging method does not require a large amount of labeled training data. In addition, CG-based taggers have proved to be highly accurate in POS tagging (cf. Chapter 3).
- The CG rules integrate much wider linear contexts into models than other tagging methods (cf. Chapter 3) because the linear context window is not fixed in CG. Similar to regular expression, the context window is defined flexibly, but the search space is inside the tokens in a sentence. Integrating

wide linear contexts allows the analysis of grammatical relations at a wide distance, which in the present case are topological field relations.[5]

- Similar to other rule-based taggers, CG rules are applied in a categorical manner (cf. Chapter 3). Therefore, the CG tagging method is suitable for modeling linguistic hard constraints (e.g. morphological agreement), whose violation causes ungrammaticality. However, it is difficult to integrate linguistic soft constraints (e.g. word order, definiteness, animacy, and information structure) into the model. Soft constraints can be better handled in a statistical manner, which is discussed in detail in the next chapter.

In the development of a CG-based supertagger, the main task comprises of the writing and processing of grammar. Grammar writing concerns the formulation of CG rules based on linguistic observations, analyses, and theories. In grammar processing, the formulated CG rules are processed by a CG compiler for debugging and testing.

In the next section, I provide the general scheme and principle of CG-based tagging. I then propose an architecture for the supertagger. I then discuss the tasks of grammar writing and grammar processing in detail.

4.2.1 Method: Constraint Grammar Tagging

CG is a reductionist approach. In CG, tagging problems lie in morphological and syntactic ambiguity. Hence, the reduction of morphological and syntactic ambiguities is regarded as the task of tagging (Karlsson et al., 1995, pp. 25ff). Accordingly, the task of the CG-based supertagger is to choose the most plausible morphosyntactic interpretations among all possible ones and to map them to syntactic tags (here: topological dependency tags). The CG-based tagger consists of three steps:

(6) Step 1: Morphological analysis (preprocessing)
Step 2: Morphosyntactic disambiguation
Step 3: Syntactic labeling (Here: Topological dependency grammar labeling)

In the first step, all possible morphosyntactic analyses are assigned to each input token. In this stage, morphosyntactic readings are provided based on word form. In my case, I use the German morphological analyzer Gertwol (Haapalainen and Majorin, 1994) for

[5] As a long distance relation refers to topicalization, scrambling and extraposition in linguistics, I use the term 'wide distance' to make reference to long distance between two tokens in general.

the morphological analysis. For example, Gertwol provides 11 morphological readings for *Der* (cf. (7)). The first column shows that the lemma and the remaining columns contain morphosyntactic features.

(7) *Der*
"der" ART DEF SG NOM MASK
"die" ART DEF SG DAT FEM
"die" ART DEF SG GEN FEM
"die" ART DEF PL GEN
"der" PRON DEM SG NOM MASK
"die" PRON DEM SG DAT FEM
"die" PRON DEM VERALTET SG GEN FEM
"die" PRON DEM VERALTET PL GEN
"der" PRON RELAT SG NOM MASK
"die" PRON RELAT SG DAT FEM
"die" PRON RELAT GESPROCHEN PL GEN

In the second step, morphosyntactic ambiguities are reduced as much as possible. This step can be divided into two subtasks (cf. Voutilainen, 1995a). The first subtask is to define contexts in which the occurrence of morphosyntactic categories is impossible. The second subtask is to define contexts in which the occurrence of morphosyntactic categories is possible. The first constraint type is the definition of the negative distribution of a morphosyntactic category (henceforth, negative distribution rules). The second constraint type is the positive distribution of a morphosyntactic category (henceforth, positive distribution rules). The former disambiguates morphosyntactic analyses passively (mildly) and safely but less effectively. The latter works aggressively and effectively but needs to be applied carefully. Technically, these two types of constraints are defined as operations, `REMOVE` and `SELECT`, respectively (cf. Appendix C). Table 4.8 provides an overview of these two types of CG rules for disambiguation.

CG rules	**CG operation**	**Selection type**	**Security/effectiveness**
Positive distribution rules	`SELECT`	Aggressive selection	Unsafe but effective
Negative distribution rules	`REMOVE`	Mild (passive) selection	Safe but not effective

Table 4.8: Constraint Grammar: Disambiguation rules

For example, the rules (9-a) and (10-a) are negative distribution rules, whereas the rules (9-b) and (10-b) are positive distribution rules. Consider that these rules are applied to disambiguate the word *Der* (cf. (7)) in the example (8). Applying a negative distribution rule is a 'mild' or 'passive' selection because all alternative morphosyntactic categories

other than the targeted one are passively selected. Thus, applying (9-a) removes the readings of the targeted relative pronouns of *Der* in the example (8). This implies that the readings of definite article (`ART DEF`) and demonstrative pronouns (`PRON DEM`) are passively selected. In contrast, applying a positive distribution rule is an 'aggressive' selection because it 'pin[s] down the correct function once and for all' (Voutilainen, 1995a, pp. 318). By applying the positive distribution rule (9-b) the readings are selected aggressively as the targeted relative pronoun of *Der*.

(8) **Der** Rückbürge ist verpflichtet, dem zahlenden Bürgen für den Rückgriff einzustehen, der diesem gegen den Hauptschuldner zusteht.

(9) a. "Relative pronoun (`PRON RELAT`) is removed if it does not follow a full noun."
b. "Relative pronoun (`PRON RELAT`) is selected if it precedes a full noun."

(10) a. "Definite article (`ART DEF`) is removed if it does not precede a full noun."
b. "Definite article (`ART DEF`) is selected if it immediately follows a full noun."

In general, because negative distribution rules are safer, they are applied before positive distribution rules. The word *Der* is correctly disambiguated as a definite article by applying the negative distribution rules before the positive distribution rules. After applying these four rules in order, the first four morphosyntactic interpretations ('cohort' in CG term) are valid and the other cohorts are marked for deletion. The semicolons indicate that the cohorts are removed (cf. (11)).

(11) *Der*

"der" ART DEF SG NOM MASK
"die" ART DEF SG DAT FEM
"die" ART DEF SG GEN FEM
"die" ART DEF PL GEN
; "der" PRON DEM SG NOM MASK
; "die" PRON DEM SG DAT FEM
; "die" PRON DEM VERALTET SG GEN FEM
; "die" PRON DEM VERALTET PL GEN
; "der" PRON RELAT SG NOM MASK
; "die" PRON RELAT SG DAT FEM
; "die" PRON RELAT GESPROCHEN PL GEN

In the third step, one or more syntactic categories are assigned to those morphosyntactic categories that have not been removed in morphosyntactic disambiguation (step 2). In our example (12), the determiner (`DET`) is assigned as a definite article by applying the constraint "definite article (`ART DEF`) is mapped to `DET`."

(12) *Der*

	"der"	ART DEF SG NOM MASK **DET**
	"die"	ART DEF SG DAT FEM **DET**
	"die"	ART DEF SG GEN FEM **DET**
	"die"	ART DEF PL GEN **DET**
;	"der"	PRON DEM SG NOM MASK
;	"die"	PRON DEM SG DAT FEM
;	"die"	PRON DEM VERALTET SG GEN FEM
;	"die"	PRON DEM VERALTET PL GEN
;	"der"	PRON RELAT SG NOM MASK
;	"die"	PRON RELAT SG DAT FEM
;	"die"	PRON RELAT GESPROCHEN PL GEN

In summary, the three steps of the CG-based tagger are as follows: 1) morphological analysis, 2) morphosyntactic disambiguation, and 3) syntactic labeling. In particular, I described two types of disambiguation rules (i.e. negative distribution rules and positive distribution rules). In disambiguation, negative distribution rules are applied before positive ones. Based on these general principles of CG, I propose an order for the CG rules in the topological dependency grammar tagging in the next section.

4.2.2 Architecture

The CG rules consist of two sets of rules: CG rules for (i) topological field tagging and (ii) dependency grammar tagging. For morphosyntactic disambiguation (cf. (4)) and syntactic tagging, I write CG rules primarily based first on linguistic hard constraints (13) and second on psycholinguistic and domain-specific preferences. In the development of the CG rules, I explore how far a rule-based supertagger can resolve morphosyntactic ambiguities by applying linguistic hard constraints.

(13)
- Topological field model
- Morphological agreement
- Argument structures (voice and valence)

The CG rules are ordered according to three general principles (14). The first and second principles are inherent in constraint grammar as I discussed in the previous section. The third principle is specific to topological dependency tagging. The morphosyntactic disambiguation required for this task is mainly at the level of coarse morphosyntactic ambiguity. The disambiguation of fine-grained ambiguity is based on the disambiguation of coarse ambiguity (cf. Section 4.1). Therefore, coarse morphosyntactic categories are disambiguated before fine-grained ones are.

(14) a. *Apply disambiguation rules at first and then labeling rules*
b. *Apply negative distribution rules at first and then positive distribution rules*
c. *Disambiguate from coarse to fine-grained morphosyntactic categories.*

Based on these principles (14), I propose an architecture for the CG-based supertagger, which is illustrated in Table 4.9. In this architecture, the morphosyntactic ambiguities are reduced, and syntactic categories are assigned incrementally in three steps. The first step is the preprocessing for topological field tagging and dependency grammar tagging. The preprocessing begins with the coarsest morphosyntactic disambiguation, that is, the one between the lexical and functional categories. Then the supertagger operates according to negative distribution rules for coarse morphosyntactic disambiguation. The topological field tagging begins after the preprocessing. In this step, topological field tags are assigned to clause brackets (e.g. verbs) by the aggressive selection of morphosyntactic categories according to the topological field model (cf. Chapter 2). The third step is dependency grammar tagging. In this step, the remaining coarse morphosyntactic categories are aggressively selected by applying the principles of morphological agreement. The case features of nouns and pronouns are further disambiguated as much as possible by applying subject-verb agreement and argument structure in order to assign dependency grammar labels to the tokens.

Step 1: Preprocessing
1.A Morphosyntactic disambiguation between lexical and functional categories (e.g. preposition vs. noun).
1.B Disambiguation of coarse morphosyntactic categories (e.g. preposition vs. determiners) by the application of negative distribution rules.
Step 2: Topological Field Tagging
Morphosyntactic disambiguation for clause brackets (e.g. verbs) by the application of positive distribution rules and the mapping of topological field tags to clause brackets (e.g. left bracket of a verb-second clause `LB-V2`).
2.A Verb-first clauses
2.B Verb-final clauses
2.C Verb-second clauses
2.D Infinitive clauses
Step 3: Dependency Grammar Tagging
3.A Disambiguation of coarse morphosyntactic categories (e.g. determiners `DET` vs. pronoun) by the application of positive distribution rules.
3.B Mapping of dependency grammar tags to adjuncts (e.g. the dependent of a preposition `PN`).
3.C Fine-grained morphosyntactic disambiguation for arguments (e.g. nominative case of nouns) by the application of negative distribution rules.
3.D Mapping of dependency grammar tags to the remaining morphosyntactic readings.

TABLE 4.9: The architecture of the rule-based topological dependency grammar tagger

In this section, I presented the architecture of the rule-based supertagger. The supertagger consists of three ordered sets of CG grammar rules. The next subsection describes the CG rules in technical detail. Defining CG rules is the core part of the development of the CG-based tagger. If the reader is not interested in the technical aspects of CG grammar writing, I suggest moving directly to the extended summary (cf. Section 4.2.3.4).

4.2.3 Grammar Writing

4.2.3.1 Step 1: Pre-Processing

In Step 1, the preprocessing takes place for both the topological field tagging (step 2) and the dependency grammar tagging (step 3). Table 4.10 provides an overview of step 1. Preprocessing resolves the ambiguity between lexical and functional categories (1.A in Table 4.10). It then mildly disambiguates coarse morphosyntactic ambiguities (1.B in Table 4.10).

Step 1: Preprocessing
1.A Morphosyntactic disambiguation between lexical and functional categories (e.g. preposition vs. noun).
1.B Disambiguation of coarse morphosyntactic categories (e.g. preposition vs. determiners) by the application of negative distribution rules.
Step 2: Topological Field Tagging
Step 3: Dependency Grammar Tagging

TABLE 4.10: Preprocessing in the architecture of the rule-based topological dependency grammar tagger

Step 1.A: Functional vs. Lexical Categories

Functional and lexical categories are vastly, sporadically, and idiosyncratically ambiguous (cf. Section 4.1.1). In law texts, functional categories usually are preferred to lexical categories (cf. (15)). For example, the word *einen* could theoretically be a verb, but it is usually used as a determiner or pronoun in the domain. The word *während* could be a participle, but it is usually used as a preposition or subordinating conjunction. *Statt* could be also analyzed as a noun,[6] but it is usually an infinitive conjunction or a preposition.[7]

Based on this observation, the ambiguity between lexical and functional categories is resolved in that lexical morphosyntactic categories are removed if they are associated with functional categories as shown in (15).[8] For example, the verbal reading of the word *einen* is removed in this way.

(15) Preference: functional category > lexical category[9]

a. Definite article > verb (e.g. *einen*)
b. Preposition > verb (e.g. *während, zulasten*)
c. Preposition > predicative adjective (e.g. *gemäss, entsprechend*)
d. Preposition > noun (e.g. *in, statt*)

[6] *Statt* can be a noun. An example is *nirgends eine bleibende Statt* (Dudenredaktion, 1999, pp. 3709)

[7] From a psycholinguistic point of view, morphosyntactic categories are lexically biased (or favored) by relative frequency (cf. Boland, 1997), which might account for the preference of functional words.

[8] There are some exceptions, such as *sein*. Such an exception is lexically handled.

[9] A > B means that A is preferred to B. In CG, the selection is written as `REMOVE verbs IF (0 determiners)`. The rule corresponds to preference (15-a) and states that all verbal morphosyntactic categories are discarded if they are alternatively analyzed as determiners.

```
REMOVE (*) IF (0 capitalized) (NOT 0 (N)) (NOT 0 truncation) (NOT 0 (A INV))
(NOT 0 first_token) (NOT -1 colon) (NOT -1 semicolon)
```

TABLE 4.11: Example constraint grammar code for the disambiguation of nouns

Step 1.B: Coarse Morphosyntactic Disambiguation

The previous step resolved the coarsest disambiguation. In this step, the less coarse morphosyntactic categories are mildly disambiguated. In this step, the preprocessing takes place before the aggressive selection of morphosyntactic categories in step 2 and step 3. The disambiguation targeted in this step is summarized in (16).

(16) a. Noun vs. verb
b. Adjective vs. verb
c. Determiner vs. pronouns and preposition vs. complementizer/conjunctions

The following subsections describe each of these competitions.

Nouns Nouns compete with the verbs (e.g. *Gleiche* in a sentence (17)[10]). In disambiguation, all morphosyntactic categories of capitalized tokens are removed if they are neither nouns,[11] indeclinable adjectives, such as *Zürcher*[12] nor positioned at the beginning of sentences[13] (cf. Table 4.11).[14] By applying this constraint, the word *Gleiche* in the sentence (17) is disambiguated as a noun, which prevents the token from being selected as a verb in the topological field tagging (step 2). It is worth noting that the tokens at the beginning of sentences are not disambiguated in this step. For example, *Verfahren* in the sentence (18-a) and *Fahren* in the sentence (18-b) are not disambiguated between nouns and verbs. For disambiguation, *Fahren* is aggressively selected as a verb in the next step; thereafter *Verfahren* is regarded as a noun.

[10]Gertwol returns the morphosyntactic analyses of nouns, adjectives, and verbs because it analyzes the forms of words regardless of their position in a sentence.
[11]There are some exceptions of the capitalization of full nouns: proper names such as *fedpol, armasuisse*
[12]Indeclinable adjectives are geographical adjectives and used as a part of proper names (e.g. *neue Zürcher Zeitung*, *die Schweizer Post*).
[13]That includes the first tokens of sentences and the tokens after semicolons or colons.
[14]The rule states that all morphosyntactic analyses (i.e. star mark *) are removed if they are capitalized. The exceptions are full nouns (`N`), truncations (`TRUNC`), indeclinable adjectives (`A INV`), and the tokens at the beginning of sentences or before colons or semicolons.

```
In NP: adjective > verb
REMOVE (V) IF (0 (A)) (1* (N) BARRIER np_stop)
```

TABLE 4.12: Example constraint grammar code for the disambiguation of adjectives

(17) Das **Gleiche** gilt für die Beschränkung und den Rückzug von Rechtsmitteln sowie für die Umwandlung von Berufungen in Anschlussberufungen.

(18) a. **Verfahren** der vorübergehenden Verwendung in der Schweiz wie das Carnet ATA stellen keine Zollveranlagung dar.

b. **Fahren** zwei Motorschiffe, von denen keines nach Artikel 44 ausweichpflichtig ist, so auf kreuzenden Kursen, dass die Gefahr eines Zusammenstosses nicht auszuschliessen ist, weicht das Schiff aus, welches das andere an Steuerbord hat.

Adjectives Adjectives compete with verbs. For example, *nahen* in *mit einer* ***nahen*** *und erheblichen Gefahr* or *bestimmten* in *bei* ***bestimmten*** *Veränderungen* are morphosyntactically ambiguous between adjectives and verbs. For disambiguation, verbs are removed if they are placed near full nouns, that is, in an NP (cf. Table 4.12).[15]

Determiners and Prepositions Determiners and prepositions compete with other functional categories including clause brackets, such as relative pronouns, subordinating conjunctions, and infinitive conjunctions (cf. Appendix F). In this step, determiners and prepositions are removed if they do not agree morphologically. The disagreement constraints (cf. Table 4.13 and 4.14) are based on the negative context in which determiners and prepositions cannot occur. Specifically, a determiner is removed if it disagrees with (i) a noun or a weakly declined adjective (`N_WeakADJ`) in number, gender, and case in an NP or (ii) a preposition in a PP (cf. Table 4.13 for case disambiguation).[16] In contrast, a preposition is discarded if its required case is not unified with the case of the determiner

[15] NP is defined in terms of a stopping criterion (i.e. `BARRIER np_stop`). The targeted cohort is regarded as within an NP if a noun (`N`) is found at the right side (`1* (N)`) in the search, but not over the stopping criterion (`BARRIER np_stop`). The stopping criterion of an NP (`np_stop`) is the set of the morphosyntactic features that cannot be placed in NP, that is, the set of all the morphosyntactic features except the following categories: adverbs, attributive adjectives, coordinating conjunctions, numbers, truncations, special characters for emphasis (e.g. "), and commas.

[16] In CG, the unification of morphosyntactic features is defined as the unification operator +. The constraint states that a determiner is removed if it is in an NP and does not agree in `case` (defined as `NOM, AKK, DAT, GEN`) with nouns or weakly declined adjectives (`N_WeakADJ`).

```
REMOVE (DET) + $$case IF (1* (N_WeakADJ) BARRIER np_stop)
(NOT 1* (N_WeakADJ) + $$case BARRIER np_stop)
```

TABLE 4.13: Example constraint grammar code for the mild case disambiguation of determiners

```
REMOVE (pre PRÄP) + $$case IF (1 (DET_StrongADJ))
(1 (DET_StrongADJ) + $$case)
```

TABLE 4.14: Example constraint grammar code for the mild case disambiguation of prepositions

or the strongly declined adjective in a PP (cf. Table 4.14[17]). For example, by applying the constraint illustrated in Table 4.14, the prepositional reading of the word *während* in the sentence (19) is discarded but not in the sentence (20). The former disagrees in PP but the latter does not. It is worth noting that not all prepositions that agree with the dependents are indeed prepositions. For example, the word *während* in the sentence (21) agrees with the determiner *der*. However, it is not a preposition, but a subordinating conjunction. In this step, the disambiguation is conducted mildly so that the case, such as *während* in the sentence (21), is still available in the processing.

(19) Wird der ausstehende Betrag ganz oder teilweise bezahlt, **während** *das* Gesuch bei der Erlassbehörde hängig ist, so fragt diese die steuerpflichtige Person an, ob sie an einer materiellen Beurteilung des Gesuches festhalte.

(20) Grenzgängerinnen und Grenzgänger, die sich **während** *der* Woche in der Schweiz aufhalten, haben sich bei der an ihrem Aufenthaltsort zuständigen Behörde anzumelden.

(21) Als Arbeitszeit im Sinne des Gesetzes gilt die Zeit, **während** *der* sich der Arbeitnehmer oder die Arbeitnehmerin zur Verfügung des Arbeitgebers zu halten hat; der Weg zu und von der Arbeit gilt nicht als Arbeitszeit.

In summary, the coarsest ambiguity, i.e. the one between the lexical and functional categories is resolved in this step. In addition, the ambiguity of coarse morphosyntactic categories is mildly disambiguated as preprocessing for step 2 and 3. In the next section,

[17] The constraint states that prepositions (`pre PRÄP`) are removed if they are adjacent to determiners or strongly declined adjectives (`DET_StrongADJ`) and do not agree with them in case.

topological field relations are assigned to clause brackets by aggressively selecting them morphosyntactically.

4.2.3.2 Step 2: Topological Field Tagging

The topological field tags that are labeled in this step are shown in Table 4.15.[18] Topological field tags are assigned to the morphosyntactic categories of clause brackets. They are coarse functional morphosyntactic categories and verbs with the distinction of finiteness (cf. Table 4.15). These morphosyntactic categories are used as possible candidates to which a topological field tag is assigned: if the conditions for the linear order of tokens match, the morphosyntactic categories of clause brackets are selected aggressively, and they are assigned simultaneously to topological field tags.

Topological Field Tag	**Clause brackets**	**Morphological Category**
`LB-V1`	Left bracket of a verb-first clause	FIN
`RB-V1`	Right bracket of a verb-second clause	NONFIN/∅
`LB-V2`	Left bracket of a verb-second clause	FIN
`RB-V2`	Right bracket of a verb-second clause	NONFIN/∅
`LB-VF-CONJ`	Left bracket of a verb-final clause (subordinated clause)	CONJ/COMP
`RB-VF-CONJ`	Right bracket of a verb-final clause (subordinated clause)	FIN
`DEP=LB-VF-RC`	Left bracket of a verb-final clause (relative clause)	RC-PRON
`RB-VF-RC`	Right bracket of a verb-final clause (relative clause)	FIN
`VC`	Part of a verbal complex	NONFIN
`TOP`	Topicalized right-most verb in a verb-second clause	NONFIN
`LXCOMP`	Left bracket of an infinitive clause	ICONJ/PUNCT
`RXCOMP`	Right bracket of an infinitive clause	ZU-INF

TABLE 4.15: Topological field tag set for supertagger

During processing, syntactic labeling is carried out incrementally in the following order of clause types according to 'simple-first' principle: 1) verb-first clauses; 2) verb-final clauses; and 3) verb-second clauses. Verb-first clauses are contextually the safest because they occur only at the beginning of sentences. Compared with verb-second clauses, verb-final clauses are more restricted than verb-second clauses because they require the left bracket and the right bracket to be occupied, which justifies the order of the processing. In each clause type, the left brackets are recognized before the right brackets are. When the topological field tagging is complete, the remaining morphosyntactic categories of clause brackets are regarded as not selected and are discarded.

[18] This table is the same as Table 2.9 in Chapter 2. I show this table here for the reader's convenience.

Table 4.16 provides an overview of the topological field tagging. This subsection is organized according to the order of processing. The processing of nonfinite clauses is described at the end of the subsection.

Step 1: Preprocessing
Step 2: Topological Field Tagging Morphosyntactic disambiguation for clause brackets (e.g. verbs) by the application of positive distribution rules and the mapping of topological field tags to clause brackets (e.g. left bracket of a verb-second clause `LB-V2`). 2.A Verb-first clauses 2.B Verb-final clauses 2.C Verb-second clauses 2.D Infinitive clauses
Step 3: Dependency Grammar Tagging

TABLE 4.16: Topological dependency grammar tagger in the architecture of the rule-based topological dependency grammar tagger

Step 2.A: Verb-First Clauses

Verb-first clauses are interrogative, imperative, and conditional clauses. In law texts, conditional V1 clauses occur frequently, whereas interrogative and imperative verb-first clauses occur rarely. Interrogative and imperative sentences are used in text types that are narrative or colloquial, whereas law texts are normative. Verb-first conditionals are positioned at the beginning of sentences and are placed in the vorfeld (e.g. sentence (22)) or in the vor-vorfeld (pre-vorfeld) (e.g. sentence (23) and (24)).[19] Verb-first conditionals in the vor-vorfeld are referred to by resultative pronouns *so* in the vorfeld (e.g. sentence (23)). In a verb-first clause, `LB-V1` is a finite verb (here: *Bejahen, Wird, Kann*) and `RB-V1` remains empty (e.g. sentence (22)) or is occupied by a nonfinite verb (here: *erhoben*). The right bracket is accompanied optionally by another verb (here: *beigelegt*), which is tagged as verbal complex `VC`.

Based on these regularities, the distribution of verbal components in verb-first conditional clauses is described in the linear pattern (25).[20]

[19] Vor-vorfeld is the sub-field before vorfeld.

[20] The patterns are formulated in regular expression for understandability. In the pattern, the `LB-V1` is occupied by a finite verb (`FIN`). It is the first token of sentences (^) or is adjacent to a punctuation mark or a coordinating conjunction (`CoordCONJ`). `RB-V1` has a nonfinite verb or remains empty (`RB-V1:NOT-FIN?`). Adjacent to `RB-V1`, the remaining verbal components are either placed as `VC` or remain absent. `LB-V2` is adjacent to a comma or a comma and a resultative adverb *so*. The commas are the boundaries of sub-fields. Some tokens are placed between the comma and `LB-V1`, excluding semicolons and colons ([^;:]*) .

(22) *Bejahen*_LB-V1 die Verlobten die Frage, wird die Ehe durch ihre beidseitige Zustimmung als geschlossen erklärt.

(23) *Wird*_LB-V1 eine allgemeine Anregung zum Beschluss *erhoben*_RB-V1, **so** ist der Gemeindeversammlung innert Jahresfrist die ausgearbeitete Vorlage zu unterbreiten.

(24) *Kann*_LB-V1 die Streitigkeit nicht durch einen Vergleich *beigelegt*_VC *werden*_RB-V1, so kann die Partei, die es wünscht, die Streitigkeit der zuständigen richterlichen Behörde unterbreiten.

(25)
```
 [comma semicolon colon CoordCONJ ^]
LB-V1:FIN [^;:]* VC:NOT-FIN* RB-V1:NOT-FIN?
comma so?  LB-V2:FIN
```

In the processing of verb-first clauses (LB-V1, RB-V1 and VC), morphosyntactic disambiguation and syntactic labeling are carried out by aggressively assigning tags to the morphosyntactic features of verbs based on the linear morphosyntactic pattern (25). Whereas LB-V1 is processed solely by the use of the pattern (25), RB-V1 and VC are recognized based on the adherence to the pattern (25) and the compatibility requirement of morphosyntactic features between two verbs (26). This morphosyntactic compatibility (26) constrains the selection of the verbal morphosyntax. The compatibility requirement is directed from a verb to another verb: A modal verb, for example, requires an infinite verb, whereas an auxiliary verb requires a zu-infinitive, infinitive, or past participle.

(26) *Morphosyntactic compatibility: Head verb → dependent verb*:

a. Modal verb → infinite verb
b. Auxiliary verb (*werden*) → infinite verb, participle past or predicative adjective
c. Auxiliary verb (*haben* and *sein*) → zu-infinite verb, past participle or predicative adjective
d. Lexical verb → verbal prefix

To apply this compatibility rule (26), the verbs are ranked according to their clause type and linear position. In a verb-first clause, the highest ranked verb (NR1 in (27)) is positioned at the left-most clause bracket position (i.e. LB-V1). The second highest verb (NR2 in (27)) holds the right-most clause bracket position (i.e. RB-V1) in a clause. The lower ranked verbs (NR3 and NR4 in (27)) are placed in less prominent positions and

are distinguished as VC. In the labeling, verbs are labeled incrementally from the highest ranked verbal position (LB) over RB to the lowest ranked verbal position VC. For example, three verbs in the sentence (24) are disambiguated in the order of *Kann*, *werden*, *belegt*, by the matching of the morphosyntactic compatibility (26-a), (26-b), respectively.

(27) *Verb ranking in verb-final clauses:*
NR1:LB-V1 ... NR4:VC NR3:VC NR2:RB-V1

Step 2.B: Verb-Final Clauses

Verb-final clauses are subordinated clauses and relative clauses. In verb-final clauses, all verbal components are adjacent and placed in the right-most positions in a clause. Subordinated clauses and relative clauses differ in their occupation of left brackets LB. In a subordinated clause, LB is a complementizer (e.g. *dass* in the sentence (28)) or a subordinating conjunction (e.g. *wenn* in the sentence (29) or (30)). In a relative clause, LB is a relative pronoun (e.g. *dem* in the sentence (31)) or a free relative pronoun (e.g. *wer*)).

(28) Die Kantone können vorsehen, *dass*_LB-VF-CONJ Neuzugezogene das Stimmrecht in kantonalen und kommunalen Angelegenheiten erst nach einer Wartefrist von höchstens drei Monaten nach der Niederlassung *ausüben*_VC *dürfen*_RB-VF-CONJ.

(29) Vor Ablauf der zweijährigen Frist kann ein Ehegatte die Scheidung verlangen, *wenn*_LB-VF-CONJ ihm die Fortsetzung der Ehe aus schwerwiegenden Gründen, *die*_LB-VF-RC ihm nicht *zuzurechnen*_VC *sind*_RB-VF-RC, nicht *zugemutet*_VC *werden*_VC *kann*_RB-VF-CONJ.

(30) Hat jemand eine fremde Sache verarbeitet oder umgebildet, so gehört die neue Sache, *wenn*_LB-VF-CONJ die Arbeit *kostbarer*_VC ist_RB-VF-CONJ *als* der Stoff, dem Verarbeiter, andernfalls dem Eigentümer des Stoffes.

(31) Während der Dauer einer Unterdeckung kann er auf den Zinssatz, mit *dem*_LB-VF-RC die Altersguthaben *verzinst*_VC *werden*_RB-VF-RC, herabgesetzt werden.

Based on these regularities, subordinated clauses (VF-CONJ) and relative clauses (VF-RC) are recognized using the patterns (32) and (33), respectively.[21] In particular, in law texts, prepositional phrases and comparative noun phrases are likely to be extraposed and moved to the nachfeld (cf. Example (30)). The extraposition reflects the pattern (33) in that a preposition (PREP) and a comparative conjunction (ComparaCONJ e.g. *als, wie*) can be placed adjacent to RB.

(32) VF-CONJ:

```
[comma semicolon colon CoordCONJ ^] ADV?
LB-VF-CONJ:SubCONJ/COMP [^:;]*?  VC:NOT-FIN* RB-VF-CONJ:FIN
[comma semicolon colon period CoordCONJ PREP ComparaCONJ]
```

(33) VF-RC:

```
N? .* [comma semicolon colon CoordCONJ] PREP?
LB-VF-RC:REL-PRON [^:; LB-V1 LB-VF-CONJ free_relative_pronouns]*
VC:NOT-FIN* RB-VF-RC:FIN
[comma semicolon colon period CoordCONJ PREP ComparaCONJ]
```

To label the left brackets of relative clauses, a further restriction is required because relative pronouns are highly ambiguous between determiners and pronouns (cf. Appendix F). In the linear morphosyntactic pattern of relative clauses (cf. (33)), the 'embeddedness' ranking (34) is also integrated into the pattern (33) to further restrict the distribution of relative clauses. In law texts, clause structures are particularly complex because of the frequency of VP coordination structures, embedded clauses, and NP coordination structures. In the ranking of 'embeddedness' (cf. (34)), higher ranked clauses are less likely to be embedded in lower ranked clauses. Equally ranked clauses (i.e. <> in (34)) are less likely to be embedded in each other. According to 'embeddedness' ranking, relative clauses are not likely to embed verb-first clauses, subordinated clauses, or free relative clauses (VF-free-RC). This restriction reflects the pattern (33) that verb-first clauses (LB-V1), subordinated clauses (LB-VF-CONJ) or free relative pronouns (LB-VF-free-RC) are used as a stopping criterion between the left brackets (LB-VF-RC) and the right brackets (RB-VF-RC) of relative clauses.

[21] Specifically, in a subordinated clause, LB occurs under two circumstances: 1) LB is the first token in a sentence (^ means the beginning of sentences); 2) LB is preceded by a punctuation mark and a coordinating conjunction (CoordCONJ). An adverb can be placed before LB (e.g. *Nur wenn*). After RB, a punctuation mark or coordinating conjunction occurs. In relative clauses, relative pronouns (REL-PRON) are preceded by punctuation marks or coordinating conjunctions. Alternatively, they follow prepositions (PREP). Although relative pronouns have antecedents, free relative pronouns do not have antecedents (cf. N? in the pattern (33)).

(34) *Principle of 'embeddedness'*

`V2 > V1 <> VF-CONJ <> VF-free-RC > VF-RC > XCOMP`

During processing, `LB` is labeled before `RB` and `VC`. `LB-VF-CONJ` and `LB-VF-RC` are aggressively labeled if they matched the pattern (32) and (33), respectively. `RB` is labeled by using the position ranking of verbs (cf. (35))[22] and the morphosyntactic compatibility between the ranked verbs (cf. (26)). In a verb-final clause, the most prominent position is the right-most clause bracket (i.e. NR1:RB-VF in (35)). It is occupied by a finite verb. The immediately lower ranked verbs are adjacent and `VC`. The morphosyntactic features of `VC` are labeled according to the morphosyntactic compatibilities illustrated in (26).

(35) *Verb ranking in verb-final clauses:*

`LB-VF ... NR4:VC NR3:VC NR2:VC NR1:RB-VF`

Step 2.C: Verb-Second Clauses

Verb-second clauses are main clauses. The sentences (36) and (37) illustrate typical verb-second clauses. In a verb-second clause, the left bracket (`LB`) (here: *kann* and *gelten*) has a finite verb. `RB` (here: *werden*) and `VC` (here: *erklärt*) are occupied by non-finite verbs or remain empty (in the sentence (37)). The vorfeld is placed by a constituent such as a finite clause (e.g. sentence (37)) or phrase (e.g. sentence (36)). In a verb-second clause, a topicalized VP is handled separately as `TOP`. The sentence (38) is a typical example of VP topicalization. `TOP` is assigned to a nonfinite verb placed at the right-most position of the VP, that is, next to the left bracket. `TOP` is positioned at the *left* side of `LB-V2`, which differentiates it from `RB-V2`.

(36) Die vor der Zivilstandsbeamtin oder dem Zivilstandsbeamten geschlossene Ehe *kann*_`LB-V2` nur aus einem in diesem Abschnitt vorgesehenen Grund für ungültig *erklärt*_`VC` *werden*_`RB-V2` .

[22] If there are more than three verbs, their positions can be alternated, as *hätte verhindern können* in the sentence (i) shows. In this construction, the highest ranked verb (here: *hätte*) is placed at the left-most position, instead of the right-most position. The remaining verbs follow the same ranking scale as (35). In this construction, the second highest ranked verb (here:*können*) is required to be infinitive.

(i) Erscheint eine ausreisepflichtige Person nicht zum Ausreisetermin, so stellt das BFM dem Kanton die Flugannullierungskosten in Rechnung, falls der Kanton die Annullierung **hätte verhindern können**.

Because this type of word order is rare, I do not discuss it here.

(37) Sobald die Überschussanteile den einzelnen Versicherungsnehmern und Versicherungsnehmerinnen zugeteilt sind, *gelten*_LB-V2 sie als geschuldet.

(38) Ohne tierärztliche Anweisung eingesetzt_VC werden_TOP dürfen_LB-V2, unter Vorbehalt der heilmittelrechtlichen Gesetzgebung, betäubende Substanzen bei nicht unmittelbar zum Verzehr vorgesehenen Fischen zur Gewinnung von Fortpflanzungsprodukten und zur Markierung oder anderweitigen Kennzeichnung sowie zur Betäubung und Tötung von Aquarienfischen.

Based on these regularities, verb-second clauses have two linear patterns (39) and (40) for RB and TOP, respectively.[23] The vorfeld constituent in the pattern (39) is specified in patterns (39–i),(39–ii) and (39–iii). They correspond to a V1 conditional clause, a VF clause, or a verbless phrase, respectively.

(39) *V2 with RB-V2*:

```
VORFELD:CONSTITUENT LB-V2:FIN  .*  VC:NOT-FIN*  RB-V2:NOT-FIN?
[comma semicolon colon period CoordCONJ]


VORFELD:CONSTITUENT
    (i)   LB-V1 [^:;]* comma so?  LB-V2:FIN
    (ii)  LB-VF [^:;]* comma LB-V2:FIN
    (iii) VORFELD:[^FIN]+ LB-V2:FIN
```

(40) *V2 with RB-V2*:

```
[^FIN LB-VF :;]* VC:NOT-FIN* TOP:NOT-FIN LB-V2:FIN:MODAL/AUX
```

During processing, LB-V2 is labeled as a finite verb if it matches one of the patterns (39) and (40). RB-V2, TOP and VC are then labeled as nonfinite verbs according to the linear patterns (39) and (40), the morphosyntactic compatibility (26), and further linguistic cues. In morphosyntactic compatibility, there are two types of verb ranking in verb-final clauses (cf. (41)). In both types, LB-V2 is the highest ranked verb. In verb-second clauses with RB-V2, lower ranked verbs follow LB-V2 in the linear order (cf. (41-a)). In contrast, in topicalized VP, lower ranked verbs are positioned at the left side of LB-V2 (cf. (41-b)).

[23] Specifically, LB-V2 is a finite verb. RB-V2 is placed at the right side of LB-V2 and is adjacent to a punctuation mark or a coordinating conjunction. A topicalized VP is handled separately (cf. pattern (40)). TOP is assigned to a nonfinite verb adjacent to LB-V2. LB has either an auxiliary verb or a modal verb.

(41) *Verb ranking in verb-second clauses:*

a. *V2 with RB-V2:* NR1:LB-V2 ... NR4:VC NR3:VC NR2:RB-V2

b. *V2 with TOP:* NR4:VC NR3:VC NR2:TOP NR1:LB-V2

For the labeling of TOP or RB-V2, the linear constraints (39) and (40) and the morphosyntactic compatibility (41) need to be further enhanced by the linguistic cues: verb type and distance (cf. (42)). The cues are indicators of the preference for linking from the highest ranked verb LB-V2 to the second highest ranked verb TOP or RB-V2. In a clause, LB-V2 can be linked either to TOP or RB-V2. In the sentence (43), for example, LB-V2, *ist*, can be theoretically linked to either *Namentlich*[24] (in this case as TOP) or *erbringen* (in this case as RB-V2). The cues provide a criterion for the selection between TOP and RB-V2. The first linguistic cue (cf. (42-a)) is the ranking of verb types. The ranking states how preferably they are linked from LB-V2. In the ranking, infinite verbs are the highest preferred verb type. They are required to be linked to modal verbs such as *können, mögen* or an auxiliary verb *werden*. In contrast, zu-infinitives, past participles, and predicative adjectives are not constrained to link with other verbs. They can remain alone, and they are used as a part of nonfinite clauses (e.g. *zu veranlassen* in the sentence (44)) or as adverbs (e.g. *geschützt* in the sentence (45) and *Namentlich* in the sentence (43)). This capability makes them less weighted than infinite verbs in the ranking (42-a). Furthermore, zu-infinitives constitute modal constructions (i.e. *haben + zu-infinitive* or *sein + zu-infinitive* with auxiliary verbs at LB-V2), whereas past participles or predicative adjectives do not. Because modal constructions are frequently used in law texts, zu-infinitives are ranked higher than past participles or predicative adjectives. Past participles are more likely to bind other verbs than predicative adjectives. Therefore, the former outrank the latter. The second cue (cf. (42-b)) is the preference for the linking of TOP/RB-V2 to LB-V2 with regard to distance. A short distance from LB-V2 is preferred to a wide distance. TOP has the shortest distance from LB-V2 because TOP is adjacent to LB-V2. The mittelfeld is divided into short and wide distances. RB-V2 is regarded to be at a short distance from LB-V2 if no commas, semicolons, or colons are placed between LB-V2 and RB-V2. LB-V2 is at a wide distance if there are commas between the two, but no semicolons or colons.

(42) a. *Type ranking*:
infinite > zu-infinitive > past participle > predicative-adjective

[24] Gertwol returns a predicative adjective reading for *Namentlich*.

b. *Distance ranking*:
vorfeld > short distance in mittelfeld > wide distance in mittelfeld

(43) **Namentlich** ist es ihr verboten, Dienstleistungen im Sinne von Artikel 29 des Gesetzes für Dritte zu erbringen.

(44) Der Eigentümer hat das Recht, ohne Zustimmung des Pfandgläubigers Veränderungen am Luftfahrzeug und an dessen Zugehör vorzunehmen oder den Umtausch von Triebwerkeinheiten im Luftfahrzeugbuch **zu veranlassen**, sofern dadurch der Wert der Pfandsache nicht offenbar vermindert wird.

(45) Tiere, die nicht unmittelbar nach ihrer Ankunft geschlachtet werden, sind auf einer ausreichend grossen Fläche und **geschützt** vor extremer Witterung unterzubringen und mit Wasser zu versorgen.

The two ranking scales (42) are again ranked (cf. (46-a)) and combined to build a harmonic ranking (cf. (46-b)).[25] In law texts, the ranking (46-a) is based on the observation that verb types indicate the preference for linking from `LB-V2` more prominently than from distance, which is illustrated in the sentences (43) and (45). In both sentences, zu-infinitives (i.e. *zu versorgen, zu erbringen*) are further distanced from `LB-V2`, than the lower-ranked `TOP/RB-V2` competitors are (i.e. *geschützt, Namentlich*). However, the zu-infinitives are linked from `LB-V2`.

(46) a. Type > Distance

b. `infinite(vorfeld)` >
`infinite(mittelfeld, short)` >
`infinite(mittelfeld, wide)` >
`zu-infinite(vorfeld)` >
`zu-infinite(mittelfeld, short)` >
`zu-infinite(mittelfeld, wide)` >
`past-participle(vorfeld)` >
`past-participle (mittelfeld, short)` >
`past-participle(mittelfeld, wide)` >
`predicative-adjective (vorfeld)` >
`predicative-adjective(mittelfeld, short)` >
`predicative-adjective(mittelfeld, wide)`

[25] The harmonic ranking is inspired by optimality theory (Prince and Smolensky, 2004)

Step 2.D: Nonfinite Clauses

Nonfinite clauses (XCOMP) are analyzed as `LXCOMP` and `RXCOMP` in the topological field model. `LXCOMP` is the left-most token in a nonfinite clause, whereas `RXCOMP` is the right-most token. Nonfinite clauses are categorized into two types: those with infinitive conjunctions (Type-A) and those without them (Type-B). Type-A nonfinite clauses are adverbial clauses (e.g. (47-a)). In this type of nonfinite clause, the left-most token is an infinitive conjunction (here:*ohne*). Type-B nonfinite clauses are nonfinite complements (e.g. (47-b-i)) or modifiers of nouns (e.g. (47-b-ii)). In this type of nonfinite clause, `LXCOMP` is assigned to a punctuation mark. In both `XCOMP` types, `RXCOMP` is assigned to a zu-infinitive verb (e.g. *sein* in the sentence (47-a)). Based on this observation, I create the linear patterns of Type-A and Type-B (cf. (48-a) and (48-b)).[26]

(47) a. Type-A:
Erhebt die Zollverwaltung biometrische Daten nach Absatz 3, **ohne**_`LXCOMP` ermittelnde Behörde **zu**_`VC` **sein**_`RXCOMP` , ist sie verpflichtet, die Löschungsanträge für diese Daten zu stellen, sobald dies nach den massgeblichen nichtzollrechtlichen Erlassen erforderlich ist.

b. Type-B:

(i) Die ETHL kann den Studierenden erlauben,_`LXCOMP` auf französische Leistungskontrollen auf Englisch **zu**_`VC` **antworten**_`RXCOMP`.

(ii) Die Pflicht,_`LXCOMP` die Passagier- und Warenlisten zur Verfügung **zu**_`VC` **stellen**_`RXCOMP`, endet sechs Monate nach Durchführung der Beförderung.

(48) a. Type-A XCOMP:

```
[semicolon colon CoordCONJ ^]
LXCOMP:InfCONJ [^:; LB-VF-CONJ LB-VF-free-RC FIN ]*
VC:NOT-FIN* RXCOMP:ZuINF
[comma semicolon colon period CoordCONJ]
```

b. Type-B XCOMP:

```
LXCOMP:[comma, semicolon, colon, CoordCONJ] ...
VC:NOT-FIN* RXCOMP:ZuINF
[comma semicolon colon period CoordCONJ]
```

[26]Similar to the disambiguation and labeling of LB-VF-RC, it is less probable that XCOMP embeds other clauses, which is defined as the stopping criterion of finite verbs (`FIN`), subordinated clauses (`LB-VF-CONJ`), or free relative clauses (`LB-VF-free-RC`).

During processing, disambiguation and labeling are ordered according to (49). The order (49) corresponds to the order of safety, that is, the strictness of the contexts in which syntactic categories appear. The reason is that zu-infinitives are syntactically ambiguous and can be mapped to not only `RXCOMP` but also `RB-V2/TOP`. The processing begins with the mapping of Type-A `LXCOMP` to infinitive conjunctions. Infinitive conjunctions (e.g. *ohne, um*) often are ambiguous with prepositions (cf. Appendix F). Here, they are aggressively selected if they match pattern (48-a). Next, based on Type-A `LXCOMP`, Type-A `RXCOMP` is labeled as a zu-infinitive verb. Among the three possible syntactic assignments of zu-infinitives (i.e. Type-A and Type-B `RXCOMP` and `RB-V2/TOP`), Type-A `RXCOMP` has the strongest evidence for the linking of the corresponding `LXCOMP`, that is, infinitive conjunctions. Infinitive conjunctions require zu-infinitives. In comparison, `LB-V2` do not always require zu-infinitives. Even for auxiliary verbs such as *haben* or *sein*, this is not the case because they can be used as full verbs. Type-B `LXCOMP` (i.e. punctuation mark) indicates the weakest cue for linking. Based on this observation, `RB-V2/TOP` and then Type-B `LXCOMP` are assigned to zu-infinitives. Finally, Type-B `LXCOMP` is assigned to the next comma, semicolon, or colon according to pattern (48-b).

(49) Safety-first principle: `LXCOMP` and zu-infinitive verbs
Type-A `LXCOMP` > Type-A `RXCOMP` > `RB-V2/TOP` >
Type-B `RXCOMP` > Type-B `LXCOMP`

In summary, in this step, topological field relations are assigned by aggressively selecting the morphosyntactic categories of clause brackets. At the end of this step, the morphosyntactic categories of clause brackets are removed. In the next step, the surviving competitors of clause brackets are aggressively selected.

4.2.3.3 Step 3: Dependency Grammar Tagging

In this step, dependency grammar labels are assigned to coarse morphosyntactic categories and fine-grained morphosyntactic categories (case features) of noun and pronouns by disambiguating the morphosyntactic ambiguities. These morphosyntactic categories were partly disambiguated in step 1 and step 2. In step 3, coarse morphosyntactic categories are aggressively selected and mapped to dependency grammar labels, and the ambiguity of case features is reduced as much as possible. The case features discriminate dependency grammar relations for most arguments (cf. Section 4.1.2), but they are morphologically highly ambiguous (cf. Appendix F). Therefore, in step 3, the dependency grammar labels are mapped ambiguously to nouns and pronouns.

Morphosyntactic disambiguation is carried out by exploiting the concept of morphosyntactic agreement in NP, PP, VP, CP, and argument structures.[27] The processing of morphosyntactic disambiguation is divided into three steps depending on the width of the following context windows:

(50) (1) BaseNP and BasePP (cf. Step 3.A in Table 4.18)
(2) MaxNP and MaxPP (cf. Step 3.A in Table 4.18)
(3) CP (cf. Step 3.C in Table 4.18)

BaseNP and BasePP are phrases that do not embed another noun and preposition, respectively. The context window is the shortest. MaxNP and MaxPP are maximally projected nouns and prepositional phrases. The context window is wider than BaseNP and BasePP. The context window CP is the widest and reaches the clause boundaries.[28]

During processing, two types of dependency grammar relations of nouns and pronouns are handled separately: arguments and adjuncts. Subject and objects belong to arguments. For example, adjuncts are genitive modifiers and appositions. An overview is provided in Table 4.17. These two types of dependency relations are separated during processing because the subject-verb agreement and argument structures used for the disambiguation in the context window CP belong to the theory of arguments. To apply these linguistic theories effectively to arguments, adjuncts are identified as much as possible before the application. Nouns that are not assigned to the syntactic functions of adjuncts are regarded as candidates for arguments in the disambiguation. Thus, the mapping of dependency grammar labels to adjuncts (cf. Step 3.B in Table 4.18) is processed before the morphosyntactic disambiguation in CP (cf. Step 3.C in Table 4.18). When the three steps are complete, dependency grammar labels are mapped to morphosyntactic categories (cf. Step 3.D in Table 4.18).

Arguments	SUBJ, OBJA, OBJD, OBJG
Adjuncts	PN, APP, KON, CJ, EXPL, PRED, ZEIT, GRAD, GMOD

Table 4.17: Dependency grammar relations for arguments and adjuncts

[27] The argument structure is a 'representation of the number and type of arguments' (Kroeger, 2004, pp. 9). The number of arguments corresponds to the valence. The types of arguments are semantic roles, such as agent, patient, experiencer, recipient, and so on (ibd:9). Semantic roles are linked to grammatical functions. Agents are typically linked to subjects, whereas patients are mapped to direct objects. Recipients and experiencers are likely to be linked to indirect objects.

[28] German clauses are commonly analyzed as CP in linguistics (e.g. Sternefeld, 2006)

This subsection is organized according to the order of disambiguation and tagging (cf. Table 4.18). I mainly discuss the morphosyntactic disambiguation of full nouns.

Step 1: Preprocessing	
Step 2: Topological Field Tagging	
Step 3: Dependency Grammar Tagging	
3.A	Disambiguation of coarse morphosyntactic categories (e.g. determiners `DET` vs. pronoun) by the application of positive distribution rules based on morphological agreement ⇒ Morphosyntactic Disambiguation in BaseNP and BasePP ⇒ Morphosyntactic Disambiguation in MaxNP and MaxPP
3.B	Mapping of dependency grammar tags to adjuncts (e.g. the dependent of a preposition `PN`).
3.C	Fine-grained morphosyntactic disambiguation for arguments (e.g. nominative case of nouns) by the application of negative distribution rules based on subject-verb agreement and argument structure ⇒ Morphosyntactic Disambiguation in CP
3.D	Mapping of dependency grammar tags to the remaining morphosyntactic readings.

TABLE 4.18: Dependency grammar tagging in the architecture of the rule-based topological dependency grammar tagger

Step 3.A: Disambiguation of Coarse Morphosyntactic Categories by the Application of Positive Distribution Rules based on Morphological Agreement

Disambiguation in BaseNP/PP In this step, the features of number, gender, and case are unified in BaseNP and BasePP. The morphosyntactic markings of number, gender, and case are more distinct in the dependents of nouns, that is, determiners and adjectives, than in nouns (cf. Appendix F). The unification of number, gender, and case in BaseNP transfers the information about the case of the dependents to that of head nouns. In a BasePP, prepositions license case. The unification of case features in BasePP propagates the case restriction of prepositions to the dependents, that is, nouns and pronouns.

In this step, nouns, determiners, and prepositions in BaseNP and BasePP are aggressively selected by agreement. The operations for agreement in this step differ from those in negative distribution rules in the preprocessing stage (cf. Subsection 4.2.3.1) because morphological agreement is used as a selection criterion but not as a removal criterion. To check morphological agreement, the case, number and gender features of nouns are

```
SELECT (N) + $$case IF (-1* (DET_ATTR) + $$case BARRIER np_stop)
```

TABLE 4.19: Example constraint grammar code for the aggressive case disambiguation of nouns with determiners or adjectives

```
SELECT (N) + $$case IF (-1* (PREP) + $$case BARRIER np_stop2)
```

TABLE 4.20: Example constraint grammar code for the aggressive case disambiguation of nouns with prepositions

unified with those of the dependents in BaseNP and vice versa (cf. Table 4.19).[29] In BasePP, the case features of nouns are unified with those of prepositions and vice versa (cf. Table 4.20).[30] The morphosyntactic features of two tokens are unified, if they have the same morphosyntactic features.

For example, *der Tierhalterin*, *dem Tierhalter*, *entstehenden Aufwand*, *das Wohlergehen*, *der Tiere* are BaseNPs in the sentence (51). After this step is complete, the case features of *Tierhalterin* are reduced to dative or genitive because *der* can be only dative or genitive in the feminine singular. The word *der* is also selected as a dative or genitive determiner, which resolves the ambiguity between the determiner and the pronoun. In the sentence (52), *von der zuständigen Stelle*, *auf ausgewiesene Teuerung*, *auf andere nicht beeinflussbare Ursachen* are BasePP. For example, after disambiguation, *Stelle* is disambiguated as dative because the preposition *von* requires the dative case. *von* is also selected as a dative preposition, which resolves the ambiguity between the preposition and the proper name (e.g. *John* ***von*** *Neumann*).

(51) Sie berücksichtigt dabei den der_DET_DAT/GEN Tierhalterin_DAT/GEN oder dem_DET_DAT Tierhalter_DAT sentstehenden_ACC/DAT Aufwand_ACC/DAT und das_DET_NOM/ACC Wohlergehen_NOM/ACC der_DET_GEN Tiere_GEN.

(52) Mehrkosten können nur berücksichtigt werden, wenn sie auf von der_DET_DAT zuständigen_DAT Stelle_DAT genehmigte Projektänderungen, auf_PREP_ACC ausgewiesene_ACC Teuerung_ACC oder auf_PREP_ACC andere_ACC nicht beeinflussbare_ACC Ursachen_ACC zurückzuführen sind.

[29] The rule states that the case features (`case`) of nouns (`N`) are selected if the determiners or attributive adjectives (`DET_ATTR`) have the same case features. BaseNP is defined in the terms of the stopping criterion `np_stop`. The stopping criterion is the set of morphosyntactic features that do not occur in NP (cf. Subsection 4.2.3.1 in detail).

[30] The rule states that the case features of nouns are unified with those of prepositions (`PREP`). BasePP is defined as the stopping criterion `np_stop2`. The stopping criterion (`BARRIER`) is the set of the morphosyntactic features that do not occur in PP.

Disambiguation in MaxNP/MaxPP In this subsection, I describe the morphosyntactic disambiguation in MaxNP and MaxPP.[31] In particular, complex noun phrases and NP coordination structures are handled in MaxNP/PP.

Complex NP and PP In law texts, complex noun phrases (complex NP) occur frequently (cf. Chapter 6). Complex NPs are noun phrases that embed participle, infinitive, or adjective phrases. For example, the sentence (53) has a complex NP, that is, *die für den Vollzug von Tierschutzvorschriften zuständigen Behörden*. Here, I define as complex prepositional phrases (complex PP) such prepositional phrases that embed complex NPs. For example, the sentence (54) has a complex PP with a participle phrase, i.e. *auf vom Bundesamt für Metrologie (METAS) anerkannte Aktivitäts-Normale*.

(53) Werden strafbare vorsätzliche Verstösse gegen die Vorschriften dieses Gesetzes festgestellt, so erstatten *die für den Vollzug von Tierschutzvorschriften zuständigen Behörden* Strafanzeige.

(54) Die Messsysteme müssen dem Stand der Technik entsprechen und *auf vom Bundesamt für Metrologie (METAS) anerkannte Aktivitäts-Normale* rückverfolgbar sein.

In complex NPs and PPs, dependency relations between the head and its dependents are at a wide distance (e.g. determiner *die* and the head noun *Behörden* in the sentence (53); preposition *auf* and the dependent noun *Aktivitäts-Normale* in the sentence (54)). The morphosyntactic features of the left-most and right-most tokens are unified by identifying a complex NP and PP. For example, in the sentence (53), *die* is ambiguous between the demonstrative pronoun, determiner, and the relative pronoun. Because relative pronouns are already processed in the topological field tagging (cf. Subsection 4.2.3.2), the choice is between the demonstrative pronoun and the determiner. The contexts in which demonstrative pronouns and the determiners of complex NP occur are syntactically similar but are used differently depending on the text type. Demonstrative pronouns are typically used in colloquial speech (e.g. *die* in the sentence (55)). In law texts, demonstrative pronouns *der, die, das* rarely occur. Therefore, determiners are aggressively selected if they follow the linear patterns of complex NP (cf. pattern (56) and (57)). Otherwise, demonstrative pronouns are selected. The morphosyntactic features of the right-most nouns are reduced by disambiguating the left-most token of a complex NP.

[31] Relative clauses can be embedded in NP and PP. However, they are handled as CP and excluded from the disambiguation of MaxNP and MaxPP.

For instance, the case features of *Behörden* in the sentence (53) are reduced to the nominative and accusative by unifying the morphosyntactic features with those of the left-most token (here: *die*).

(55) Die Tante? *Die* habe ich schon lange nicht mehr gesehen. (self-created)

Complex NP and PP are recognized by using linear patterns (56) and (57), respectively. The pattern of a complex NP (cf. pattern (56)) is based on the observation that the left-most token, i.e. determiner (`DET`) or attributive adjective (`ADJ`), is followed by a cue, such as pronominal adverbs (`PRONADV`), pronouns (`PRON`), determiners (`DP`), prepositions (`PREP`), or comparative conjunctions (`ComparaNP`) without being interrupted by morphosyntactic features except NP elements. The NP elements are adverbs (`ADV`), attributive adjectives (`ADJ`), numbers (`NR`), coordinating conjunctions (`CoordCONJ`), and commas (`COMMA`). In the pattern of a complex PP (57), prepositions (i.e. preposition `PREP` or `PRONADV`) are placed sequentially without separation by tokens except the NP elements. In the patterns (56) and (57), the cues are then followed by attributive adjectives (`ADJ`), attributive present participles (`PART-PRES`), or zu-infinitives (`ZU-INF`). Table 4.21 provides examples that can be recognized by using patterns (56) and (57).

(56) Complex NP:

```
LEFT-EDGE:[DET ADJ] [ADV ADJ NR CoordCONJ COMMA]*
CUE:[PRONADV PRON DP PP ComparaNP] [ADV ADJ NR CoordCONJ COMMA]*
PRED:[ADJ ZU-INF PART-PRES] [ADV ADJ NR CoordCONJ COMMA]*
RIGHT-EDGE:[N]
```

(57) Complex PP:

```
LEFT-EDGE:[PREP] [ADV ADJ NR CoordCONJ COMMA]*
CUE:[PRONADV PREP] [ADV ADJ NR CoordCONJ COMMA]*
PRED:[ADJ ZU-INF PART-PRES] [ADV ADJ NR CoordCONJ COMMA]*
RIGHT-EDGE:[N]
```

Types	**Cues**	**Examples**
Complex NP	PRONADV	*den damit verbundenen Stress*
Complex NP	PRON	*die ihr gewährte Exportrisikoversicherung*
Complex NP	DP	*ein den Tieren angepasstes Klima*
Complex NP	PP	*einen eintägigen, durch das BVET veranstalteten Einführungskurs*
Complex NP	ComparaNP	*der formrichtig als Papier-Schuldbrief erstellte Pfandtitel*
Complex PP	PRONADV	*im daran anschliessenden Masterstudium*
Complex PP	PREP	*in am 1. September 2008 bestehenden Räumen*

TABLE 4.21: Examples for complex noun phrases and complex prepositional phrases using linear patterns (56) and (57)

As determiners, pronouns and prepositions are selected in this step, the disambiguation of all morphosyntactic categories except case features of nouns and pronouns are now completed.

Coordinated NP and PP Complex coordinated noun phrases occur frequently in law texts. The sentence (58) is a typical example in which men and women are treated linguistically equal. In the sentence (58), four professions are formulated with a pair of a woman and a man: *Die Herstellerin oder der Hersteller*, *die Importeurin oder der Importeur*, *die Verkäuferin oder der Verkäufer* and *der Tierhalterin oder dem Tierhalter*.

(58) Die Herstellerin oder der Hersteller, die Importeurin oder der Importeur, die Verkäuferin oder der Verkäufer muss die mit der Bewilligung verbundenen Bedingungen und Auflagen der Tierhalterin oder dem Tierhalter spätestens bei Auftragsannahme schriftlich bekannt geben.

In coordinated NPs, coordinated nouns agree in case. By identifying coordinated noun phrases, the case features of nouns can be further reduced. For example, *Herstellerin* in the sentence (58) is ambiguous between the nominative and the accusative. By identifying the coordination structure (here: *Die Herstellerin oder der Hersteller*), *Herstellerin* is disambiguated as the nominative.

Three cues are employed to identify coordinated noun phrases: case agreement, distance, and determiner type (cf. (60)). The cues are used to disambiguate the ambiguity in the attachment of coordination between two nouns. For instance, the coordinated NP *Auflagen der Tierhalterin oder dem Tierhalter* in the sentence (58) is ambiguous in the attachment structure (cf. (59)). In the attachment (59-a), the underlined nouns *Tierhalterin* and *Tierhalter* are coordinated, whereas *Auflagen* and *Tierhalter* are coordinated

in (59-b). Morphosyntactically, *Tierhalterin* is dative in (59-a) and genitive in (59-b). The first cue (cf. (60-a)), case agreement, is the compatibility of the case features of coordinated nouns. The cue is divided into two groups: agree or disagree. Case agreement is a grammar constraint that must be fulfilled. For example, *Finanzmarktaufsicht* in the sentence (61) can be attached to either *Statistik* or *Bundesamt*. Examining the compatibility of case features reveals that *Finanzmarktaufsicht* cannot be attached to *Statistik*. In the distance cue (cf. (60-b)), two types of attachment structures are defined depending on the distance between coordinated nouns: early attachment and late attachment. In early attachment, coordinated nouns are attached as early as possible. For example, the attachment structure (59-a) is an example of early attachment. In late attachment, coordinated nouns are defined as the next best nouns that agree in case. For example, the nouns in the attachment structure (59-b) are coordinated in late attachment. Early attachment is favored over late attachment according to a psycholinguistic strategy of human parsing: late closure. In the late closure strategy, "when possible, attach incoming material into the clause or phrase currently being parsed" (Frazier, 1979, pp. 76).[32] In late closure, humans parse sentences with as little memory as possible. The third constraint (cf. (60-c)), determiner type, is based on the observation that nouns with the same type of determiners are more likely to be attached. As determiner types, we consider *der, ein, kein, dieser, jener, meiner* and none (bare or adjective). The coordination structures in the sentence (61) shows that the definiteness type of nouns plays an important role in the attachment of coordination structures. While the standalone noun *Statistik* is a bare noun, the coordinated nouns *Bundesamt, Finanzmarktaufsicht, Behörden* are all accompanied by definite articles.

(59) a. Auflagen [der Tierhalterin oder dem Tierhalter]
b. [[Auflagen der Tierhalterin] oder dem Tierhalter]

(60) a. Case agreement: *Agree* > *disagree*
b. Distance: *Early attachment* > *late attachment*
c. Determiner types: *The same type of determiners* > *the different type of determiners*

[32] As closely related strategy to late closure, there are Bever's strategy A ("Segment together any sequence X ... Y, in which the members could be related by primary internal structural relations: actor-action-object ... modifier"(Bever, 1970, pp. 290)) and Kimball's right association principle ("Terminal symbols optimally associate to the lowest non-terminal node." (Kimball, 1973, pp. 24))(cf. Frazier, 1979). The late closure principle was applied previously in parsing by Hobbs and Bear (1990). Pereira (1985), Shieber (1983) proposed a shift-reduce parser according to late closure and Frazier's minimal attachment ("Attach incoming material into the phrase-marker being constructed using the fewest nodes consistent with the wellformedness rules of the language." (Frazier, 1979, pp. 76))

(61) Sie arbeitet bei der Erhebung statistischer Daten mit den zuständigen Stellen des Bundes, insbesondere mit *dem Bundesamt*_DAT *für Statistik*_AKK *und der Eidgenössischen Finanzmarktaufsicht*_GEN/DAT, *den zuständigen Behörden*_DAT *anderer Länder* und mit internationalen Organisationen zusammen.

The cues are ranked to build heuristics (cf. (62)). Morphosyntactic agreement is the condition to be fulfilled, which makes it the highest ranked cue. Determiner type is a more reliable cue than distance is (cf. Example (61)); hence, the determiner type outranks distance. This ranking is based on the observation that nouns in law texts are more likely to be coordinated if they have the same determiner type regardless of distance. The sentence (61)) is a typical example. The rankings (60) and (62) combine to constitute a new ranking (63). In combination, the disagreement of case features is excluded. As agreement in case is the condition for late attachment; it is parenthesized in (63).

(62) Morphosyntactic agreement > Determiner type > Distance

(63) The same determiner type & early-attachment & agreed in case >
The same determiner type & late-attachment (agreed in case) >
The different determiner types & early-attachment & agreed in case >
The different determiner types & late-attachment (agreed in case)

In the processing, the ranking (cf. (63)) is applied to find plausible coordinated pairs for attachment. Nouns that are regarded as candidates for coordinated nouns are either preceded by or followed by a comma, semicolon, or coordinating conjunction. Candidate nouns in early attachment are the next 'possible' upward nouns from current tokens, whereas candidate nouns in late attachment are the next 'best' upwards nouns after that.[33] The next 'best' nouns in late attachment are candidates that agree in case with the current tokens. Table 4.22 illustrates the processing of the search for candidate nouns in early and late attachment by using an example of the nouns in the sentence (61). For the current token *Behörden*, the successfully selected attachment is the token *Finanzmarktaufsicht* because it fulfills the highest ranked constraint in the ranking (63). They have the same determiner type, they are connected in early attachment, and they agree in case. In late attachment, the next best noun for the current token *Behörden* is the token *Bundesamt*. *Statistik* is the next 'possible' noun in late attachment but not the 'best' one because it does not fulfill the requirement of case agreement.

[33]The stopping criterion is the set of morphosyntactic features that cannot be placed in NP and PP.

Current token	**Candidates for attachment**		
	Bundesamt	*Statistik*	*Finanzmarktaufsicht*
Finanzmarktaufsicht	[#2] Late attach. succeeded (D, A)	[#1] Early attach.: failed (D, A)	
Behörden	[#2] Late attach.: not processed	not candidate: skipped	[#1] Early attach.: succeeded (D,A)

TABLE 4.22: Identification of attachment of coordination structures in the sentence (61): # = the order of the processing, D = determiner type, A = morphosyntactic agreement

Step 3.B: Mapping of Dependency Grammar Tags to Adjuncts

In this step, dependency grammar relations are assigned to adjunct nouns and pronouns. This step is considered the pre-processing for the next step (3.C) in which the case features of argument nouns are disambiguated.

Adjuncts are categorized into two types: the tags the cases of which are dependent on the cases of the heads (Type A in Table 4.23) and tags whose cases are inherent (Type B in Table 4.23). In Type A, dependency grammar labels are mainly related to the heads at a short distance, and they agree in case. `PN`, `APP`, `KON` and `CJ` belong to this group. Type B dependency grammar labels are constrained by their own case feature: `EXPL` and `PRED` are nominative nouns, `ZEIT` and `GRAD` are accusative nouns, and `GMOD` is genitive. For the labeling of adjunct dependency relations, I created simple CG rules based on heuristics. The following paragraphs describe the labeling process.

Adjuncts	*Type A*: PN, APP, KON, CJ
	Type B: EXPL, PRED, ZEIT, GRAD, GMOD

TABLE 4.23: Two types of dependency grammar relations for adjuncts

PN The dependents of prepositions are analyzed as `PN` in Foth's dependency grammar (cf. Chapter 3) (cf. Example (64)). To map `PN`, the heuristic (65) is applied.

(64) Die Zulage wird nur für **Milch**_`PN` ausgerichtet, die ohne **Zusatzstoffe**_`PN` gemäss **Lebensmittelgesetzgebung**_`PN` mit **Ausnahme**_`PN` von **Kulturen**_`PN`, Lab und Salz und ohne **Behandlungsmethoden**_`PN` wie Pasteurisation, Baktofugation oder andere Verfahren mit gleicher **Wirkung**_`PN` verarbeitet wurde.

(65) Assign `PN` to a noun if it is in PP.

APP In law texts, appositions (APP) are often a part of law references (e.g. *Artikel 48 Absatz 3 VRV* in the sentence (66-a)) or bare nouns adjacent to nouns (e.g. *SN* in the sentence (66-a)) or parentheses (e.g. *NDB* in the sentence (66-b)).[34] Based on this observation, two heuristics, (67-a) and (67-b), are applied to identify APP.

(66) a. Kleidung nach Artikel 48 **Absatz**_APP 3 **VRV**_APP, die nicht der Schweizer Norm **SN**_APP 640 710 entspricht, kann noch bis zum 31. Dezember 2000 verwendet werden.

b. Er regelt die Aufgabenteilung zwischen dem Nachrichtendienst des Bundes (**NDB**_APP) und dem Bundesamt für Polizei (**fedpol**_APP) sowie zwischen diesen und den Organen der militärischen Sicherheit während eines Assistenzdienstes oder eines Aktivdienstes.

(67) a. Legal reference:
Assign APP to a noun if it is a section name (e.g. *Kapitel, Teil, Abschnitt*) and precedes or follows a number or an alphanumeric word (e.g. *1, 2bis*).

b. Bare nouns:
Assign APP to a bare noun if it is preceded by a noun and followed by a number or an alphanumeric word (cf. *SN* in *Schweizer Norm SN 640 710*) or if it is preceded by an opening parenthesis (cf. *NDB* in *(NDB)*).

KON and CJ In dependency grammar, coordinated nouns are analyzed as KON or CJ. KON or CJ differ in the type of conjunctions, that is, coordination with a comma or a coordinating conjunction, respectively. For example, *Richterinnen* and *Richter* in the sentence (68-a) are mapped to KON and CJ. In addition, CJ is assigned to the dependent noun of a comparative conjunction (e.g. *Gründe* and *Nutzung* in the sentence (68-b)). Based on this observation, KON is identified by applying the heuristic (70-a) and CJ by the heuristics (70-b) and (71).

(68) a. Die Abgeordneten, die **Mitglieder**_KON der Regierung, die **Richterinnen**_KON und **Richter**_CJ, die Staatsanwältinnen_KON und **Staatsanwälte**_CJ und die **Mitglieder**_CJ der Bezirks- und der **Gemeindebehörden**_CJ, die vor dem Inkrafttreten dieser Änderung gewählt worden sind, bleiben dies bis zum Ende ihrer vierjährigen Wahlperiode.

[34] In newspapers, commas are frequently used as the boundaries of appositions, but not in law texts

b. Nicht als wichtige Gründe_CJ gelten finanzielle Interessen, wie die möglichst einträgliche Nutzung_CJ des Bodens oder die billige Beschaffung von Land für nichtforstliche Zwecke.

(69) Wer der Pflicht, den Berechtigungsausweis vorzuweisen und abstempeln zu lassen, nicht nachkommt, wird mit Busse bestraft.

(70) KON and CJ in coordination structures[35]

a. Assign KON to a noun if an NP-coordination comma is placed before the NP and precedes a noun.

b. Assign CJ to a noun if an NP-coordination conjunction is placed before the NP and precedes a noun.

(71) CJ in comparative noun phrases:
Assign CJ to a noun if a comparative conjunction is placed before the NP.

EXPL/PRED Expletives *es* are syntactically and semantically empty and are placeholders in vorfelds.[36]

The real subjects (e.g. *keine Sackgassen* in the sentence (72)) are placed in the mittelfeld. For the identification of EXPL, the heuristic (73) is applied. The mismatch of subject-verb agreement is used as a cue (cf. Example (72)).[37]

(72) **Es**_EXPL dürfen keine Sackgassen vorhanden sein.

(73) Assign EXPL to *es* if it is placed in vorfeld and does not agree with the adjacent verb.[38]

[35] NP-coordination conjunctions and NP-coordination commas are both annotated based on the recognition of NP coordination (cf. 4.2.3.3). NP-coordination conjunctions and commas are marked to distinguish them from the other types of coordination structures, such as VP-coordination and the other types of comma use (i.e. separation of phrases and clauses)

[36] The definition of expletive *es* in this work excludes impersonal *es*, such as *es regnet*, and correlative *es*, such as *Die Abgeltung wird pauschal bestimmt oder* ***es*** *wird festgelegt, welche massgebenden Kosten zu welchem Satz abgegolten werden.* These uses of *es* are semantically plastic, but they have a syntactic function in a clause.

[37] In the case that *es* is in the vorfeld and agrees with the verb, it is assumed that it is not expletive, since *es* is more likely to bear a grammatical function.

[38] Theoretically, *es* in the vorfeld could be accusative. However, it is more likely that pronouns in the vorfeld are subjects.

PRED is the complement in a copula construction (e.g. *Bundesfeiertag* in the sentence (74)). In a copula construction, SUBJ and PRED both are in the nominative case. The distinction between SUBJ and PRED is semantically conditioned. However, the subject tends to precede PRED. PRED is assigned after the disambiguation of case in CP by applying the heuristic (75).

(74) Der 1. **August**_SUBJ ist **Bundesfeiertag**_PRED.

(75) Assign PRED to a nominative noun if it follows SUBJ in a copula construction.

ZEIT/GRAD The measurement expression GRAD and the time expression ZEIT are accusative and compete with the accusative object (OBJA). GRAD denotes the unit of measurements (e.g. *Jahre* in the sentence (76-a) and *Monate* in the sentence (76-b)), and it modifies another noun pre-nominally. A heuristic (77) is applied (cf. Foth, 2005) to identify GRAD.

(76) a. Endigt das Arbeitsverhältnis eines mindestens 50 **Jahre**_GRAD *alten*_ATTR Arbeitnehmers nach 20 oder mehr Dienstjahren, so hat ihm der Arbeitgeber eine Abgangsentschädigung auszurichten.

b. Die zuständige ausstellende Behörde bewahrt das Antragsformular zwei **Monate**_GRAD *lang*_ADV auf.

(77) GRAD is assigned to an accusative noun if it follows a number and precedes an attributive, adverbial, or comparative adjective.

ZEIT denotes the time and is used as temporal adverb.[39] In tagging, three heuristics (79) are used to map ZEIT (cf. Foth, 2005):

(78) Der Importeur hat spätestens 60 **Tage**_ZEIT nach Jahresende eine Meldung über die effektiv eingeführten Mengen an Chemikalien der Listen 2 und 3 je Herkunftsland zu erstatten.

[39] There is, in some minor cases, genitive ZEIT, e.g. *eines Tages* or *eines Abends*. Genitive ZEIT refers to an unspecific time and is used in narrative texts, but not in law texts. Therefore, the genitive time expression is assigned to the genitive modifier (GMOD) in the post-processing step 3.D.

(79) a. Assign ZEIT to an accusative noun if it belongs to a countable time expression (i.e. year, month, day, hour) and if it is adjacent to a number, e.g. *24 Stunden*

b. Assign ZEIT to an accusative noun if it belongs to an uncountable time expression (i.e. week day and month) (e.g. *Montag*)

c. Assign ZEIT to an accusative *Anfang, Mitte, Ende* if it is adjacent to a time expression, e.g. *Anfang Monat*

GMOD Genitive nouns are ambiguous. They can be mapped either to the genitive object (OBJG) or to the modifier of a noun (GMOD). GMOD must have a head noun to constitute a noun phrase. A pre-nominal GMOD occurs only if GMOD is name of a person e.g. *Mozarts Wiegenlied*. Otherwise, GMOD is a post-nominal. Based on this observation, two rules (82) are applied for the mapping of GMOD. For example, by applying the first heuristic, *Besuches, Lehranstalt, Person* in the sentence (80) are identified as GMOD. The use of the second heuristic identifies *Persönlichkeit* as GMOD in the sentence (81). However, these two rules are too restrictive and do not assign all possible GMOD. GMOD is mapped to the remaining genitive nouns after the assignment of OBJG in the next step. OBJG is syntactically and semantically more restricted than GMOD is.

(80) Der Aufenthalt an einem Orte zum Zweck des **Besuches**_GEN_GMOD einer **Lehranstalt**_GEN/DAT_GMOD und die Unterbringung einer **Person**_GEN/DAT_GMOD in einer Erziehungs-, Versorgungs-, Heil- oder Strafanstalt begründen keinen Wohnsitz.

(81) Jeder Mensch hat das Recht auf freie Entfaltung seiner **Persönlichkeit**_GEN/DAT_GMOD und auf Chancengleichheit.

(82) a. Assign GMOD to genitive nouns if they are placed in a vorfeld and are not the first noun. Vorfeld is not occupied by CP or topicalized VP.

b. Assign GMOD to genitive nouns if they are placed within coordinated NP and PP.

Step 3.C: Fine-grained Morphosyntactic Disambiguation of Arguments by the Application of Negative Distribution Rules Based on Subject-Verb Agreement and Argument Structure

In this step, the case features are further disambiguated in the context window of clauses (CP). In this step, all nouns that are not assigned to adjunct dependency grammar

relations in the previous step (3.B) are regarded candidates for arguments and are disambiguated by exploiting the heuristics based on argument structure and subject-verb agreement in CP. The annotation of topological fields described in Subsection 4.2.3.2 is employed as the stopping criterion of CP.

In this section, I describe the most frequent case ambiguities that are disambiguated in CP: the disambiguation between genitive and dative nouns and between nominative and accusative nouns. These ambiguities cannot be resolved by agreement in BaseNP/PP and MaxNP/PP in step 3.A. In this step, the case ambiguity is reduced but is not forced to be resolved completely. Therefore, nouns can be tagged as having more than one dependency grammar relation.

Dative Case vs. Genitive Case Genitive and dative features often are ambiguous because of the feminine gender class. Morphosyntactically ambiguous words can be assigned to the genitive modifier (`GMOD`), genitive object (`OBJG`), and dative object (`OBJD`). In the following paragraphs, I create heuristics (cf. Table 4.24) based on the analysis of differentiating properties of three categories. In the heuristics, the complement `OBJD` and adjunct `OBJD` are conceptually differentiated. Dependency grammar does not distinguish between arguments and adjuncts for `OBJD`.[40]

Nr.	Rules
A	Only few verbs require `OBJG` and complement `OBJD` as arguments. Free `OBJD` and `GMOD` are adjuncts and appear freely regardless of the argument structure in a clause.
B	`OBJG`, complement `OBJD` and free `OBJD` can occur just once in a clause. (`GMOD` can appear more than once in a clause.)
C	Complement `OBJD` does not co-occur with `OBJG` and `OBJA` in a clause.

TABLE 4.24: Rules: GMOD, OBJG, adjunct OBJD and argument OBJD

`Rule-A` A few verbs require a complement dative object or a genitive object; I extracted 239 verbs for `OBJD` and 37 verbs for `OBJG` automatically.[41] The lists of verbs (cf. Table 4.25) serve to restrict clauses in which `OBJG` and complement `OBJD` are assigned. A further restriction is the position of argument candidates. Complement objects, `OBJD`, and `OBJG` tend to be placed in prominent positions (i.e. first noun in the vorfeld or the mittelfeld), whereas the placement of adjunct objects (free `OBJD` and `GMOD`) is freer.

[40] Unlike complement `OBJD`, 'free' or 'adjunct' `OBJD` can occur in a clause even if the predicate does not require a dative object e.g. *Mir läuft die Zeit davon* or *Den Kuchen habe ich mir selbst gebacken.*

[41] Resource from the Constraint Dependency Grammar Software: https://nats-www.informatik.uni-hamburg.de/CDG (last visited: 18/04/2015).

Based on this observation, I create a heuristic (cf. (83)). For example, by applying the heuristic, *Erklärung* in the sentence (84) is disambiguated and labeled as `OBJG`, because it is the first noun in the mittelfeld, and the predicate *bedürfen* requires a genitive object. *Ehegatten* and *Einwilligung* are not assigned as `OBJG` because they are not placed in the prominent position defined in the heuristic (83).

Arguments	Verbs
OBJD	fehlen, abraten, angehören, entfliehen, gefallen, gehorchen, gehören, gelingen, genügen, gegenübersitzen, helfen, mangeln, misslingen, misstrauen, nachkommen, nachlaufen, nahestehen, unterliegen, unterstehen, vertrauen, vorausgehen, wehtun, widerfahren, widersprechen, widerstehen, ähneln, etc.
OBJG	anklagen, anschuldigen, bedienen, bedürfen, begeben, belehren, bemächtigen, berauben, beschuldigen, besinnen, bezichtigen, erfreuen, erinnern, ermangeln, erwehren, gedenken, verdächtigen, würdigen, etc.

TABLE 4.25: Examples of OBJD and OBJG verbs

(83) Assign `OBJG` or `OBJD` to a genitive noun or a dative noun in a clause if the predicate requires `OBJG` or `OBJD`, respectively, and if it is (i) the first noun in the vorfeld or (ii) the first noun in the mittelfeld.

(84) Zu Verfügungen über Vermögenswerte des Gesamtgutes *bedarf* es einer **Erklärung**`_ARG_DAT/GEN_OBJG` der beiden **Ehegatten**`_ARG_DAT/GEN` oder der **Einwilligung**`_ARG_DAT/GEN` des einen zur Verfügung des andern, sobald es sich um mehr als die gewöhnliche Verwaltung handelt.

After the application of the rule (83), genitive readings are removed if they are placed in the prominent positions, as they can be neither genitive modifiers nor genitive objects. Applying the rule (86), the token *Behörde* in a ditransitive sentence (86) is disambiguated as dative.

(85) Remove the genitive feature of argument candidates if it is (i) the first noun in the vorfeld or (ii) the first noun in the mittelfeld.

(86) Die Pflegeeltern müssen der **Behörde**`_ARG_DAT/GEN` innerhalb von zehn Tagen die Einreise des Kindes mitteilen.

`Rule-B` `OBJG` and `OBJD` cannot appear more than once in a clause, whereas `GMOD` can modify nouns as many times as possible. Based on this observation, I create two heuristics (cf. (87)). For example, the sentence (84) has two nouns *Erklärung* and *Ehegatten*, which are ambiguous between the dative and the genitive. The token *Erklärung* is tagged

as OBJG, by exploiting Rule-A. Applying the heuristic (87-b), the tokens *Ehegatten* and *Einwilligung* are mapped to GMOD.

(87) a. Remove the dative feature of argument candidates if a OBJD is in a clause.
b. Assign GMOD to genitive argument candidates if there is a OBJG in a clause.

Rule-C Complement OBJD does not co-occur with OBJG and OBJA in a clause. Based on the restriction of the co-occurrence of a complement OBJD with other objects (OBJG and OBJA), I create two heuristics (cf. (88)). For instance, the token *Geschäftsführung* in the sentence (89) is disambiguated as the nominative by applying the first heuristic. The token *Aufsicht* is already assigned as a complement indirect object OBJD, by applying the rule (83), as the predicate *unterliegen* requires a complement dative object.

(88) a. Remove the accusative and dative features of argument candidates if there is a complement OBJD in a clause.
b. Map GMOD to genitive argument candidates if there is a complement OBJD in a clause.

(89) Die Geschäftsführung_NOM/ACC der Grundbuchämter unterliegt der administrativen **Aufsicht**_DAT/GEN_OBJD der Kantone.

Nominative Case vs. Accusative Case The nominative case and the accusative case are ambiguous if the nouns are feminine or neuter in gender or plural in number. I create heuristics to be applied based on the rules of agreement and argument structure shown in Table 4.26.

Nr.	Rules
1	Every (finite) clause has a subject.
2	Subject agrees with the verb in a (finite) clause.
3	Passive sentences, adjective clauses and copula sentences do not contain an accusative object.
4	Subject is not overt in a nonfinite clause.

TABLE 4.26: Rules: SUBJ and OBJA

RULE-1 Every clause has a subject that bears a nominative case feature, which implies two heuristics (90). Applying the heuristic (90-a) and (90-b), SUBJ is mapped to *Schiffsersatzteile* in the sentence (91-a) and *Gesuch* in the sentence (91-b), respectively.

(90) a. Assign SUBJ to an argument candidate if there is only one argument candidate in a clause.
b. Assign SUBJ to an argument candidate if there is only one nominative argument candidate in a clause.

(91) a. Nicht als Schiffsvorräte_CJ gelten **Schiffsersatzteile**_ARG_NOM/ACC/DAT/GEN und die Schiffsausrüstung_CJ.
b. Für Ehrengaben_PN ist der Zollkreisdirektion_ARG_DAT/GEN vor der Einfuhr_PN ein **Gesuch**_ARG_NOM/ACC um Zollbefreiung_PN einzureichen.

RULE-2 An accusative object does not occur in a passive sentence, an adjective clause, or a copula sentence. Based on this observation, I create the heuristic (92). Applying the heuristic, *Eingriffe* in the sentence (93) is disambiguated as nominative.

(92) Remove the accusative feature of argument candidates in a passive sentence, an adjective predicate, and a copula sentence.

(93) Nur zu Übungszwecken dürfen keine **Eingriffe**_ARG_NOM/ACC an Tieren vorgenommen werden, ausser wenn ein Eingriff nach Artikel 141 TSchV als Tierversuch bewilligt worden ist.

RULE-3 The subject agrees with the finite verb in number and person in a clause. I create a heuristic based on subject-verb agreement (cf. (94)). In the sentence (95), the nominative feature of the plural noun *Versuchstierhaltungen* is removed because *kontrolliert* is singular.

(94) Remove the nominative case feature of argument candidates if they do not agree in person and number with the finite verb in a clause.

(95) Die kantonale Fachstelle_ARG_NOM/ACC_SG kontrolliert_SG die **Versuchstierhaltungen**_ARG_NOM/ACC_PL jährlich mindestens einmal.

RULE-4 In a nonfinite clause, the subject is not overt. Based on this observation, I create a heuristic (cf. (96)). The nominative feature of *Ausbildung* is removed in the sentence (97).

(96) Remove the nominative features of argument candidates if they are in an infinitive clause.

(97) Sie kann bei Bedarf diese Personen verpflichten, eine ergänzende **Ausbildung**_ARG_NOM/ACC zu absolvieren.

Step 3.D: Mapping of Dependency Grammar Tags to the Remaining Cohorts

In the previous step, all morphosyntactic disambiguation in the rule-based tagger is completed. However, not all ambiguities in dependency grammar relations are solved. In this processing, valid cohorts that have not yet been mapped to any dependency grammar tags are labeled. The mapping rules are illustrated in Table 4.27. The nominative, accusative, dative, and genitive features of argument candidates are mapped to SUBJ, OBJA, OBJD, GMOD, respectively. The mapping of determiners, attributives, numbers, adverbs and comparative conjunctions is performed by one-to-one mapping (cf. Subsection 4.1.2). Regarding prepositions, the rule-based supertagger does not distinguish between OBJP (complement) and PP (adjunct).

Dependency tag	**Gertwol and mapping conditions**
SUBJ	NOM (noun or pronoun)
OBJA	AKK (noun or pronoun)
OBJD	DAT (noun or pronoun)
OBJG	GEN (noun or pronoun)
DET	DET and ART
ATTR	A, A(PART), NUM (inflected)
ATTR	Number if it is not APP
APP	Number if it follows a noun
ADV	ADV
KOM	sKONJ
PREP	pre PRÄP

TABLE 4.27: Mapping of dependency grammar labels from morphological analyses

4.2.3.4 Summary

In summary, this subsection described the grammar writing of three types of CG rules used in topological dependency grammar tagging. For the benefit of readers who did not read the technical details of this process, I summarize these three steps as follows:

Step 1: Preprocessing

- Step 1 is the preprocessing for topological field tagging (step 2) and dependency grammar tagging (step 3)

- In this step, the ambiguity of morphosyntactic analyses is reduced passively and mildly by discarding coarse morphosyntactic readings in negative conditions. The assignment of topological dependency grammar tags is not included in this step.
- The passive morphosyntactic disambiguation is carried out to avoid morphosyntactic categories being selected aggressively in step 2 and step 3.

Step 2: Topological field tagging

- Step 2 is the tagging of topological field relations. Morphosyntactic disambiguation and tagging are carried out simultaneously.
- In this step, topological field tags are aggressively assigned to morphosyntactic analyses belonging to clause brackets by selecting them in positive conditions. The morphosyntactic categories of clause brackets are verbs, subordinating conjunctions, infinite conjunctions, complementizers, and relative pronouns. The positive conditions are defined mainly based on the topological field model.
- In this step, only one topological field tag is assigned to a token and the morphosyntactic categories of clause brackets are completely disambiguated.

Step 3: Dependency grammar tagging

- In Step 3, dependency grammar relations are tagged. Morphosyntactic disambiguation and tagging are carried out incrementally and incompletely.
- In this step, dependency grammar tags are divided into (A) the adjunct of a predicate and (B) the argument of a predicate. The dependency grammar relations (A) are aggressively assigned to corresponding morphosyntactic features by selecting them in positive conditions. The positive conditions are defined mainly based on morphological agreement principles. Morphosyntactic features (i.e. case features of nouns and pronouns) belonging to dependency grammar relations (B) (e.g. subject) are passively and incompletely disambiguated by defining negative and positive conditions based on morphological agreement principles, including subject-verb agreement and argument structure. It is worth noting that canonical word orders are not used to define the conditions under which dependency grammar labels are assigned to arguments.
- A token can be assigned to more than one dependency grammar tag if it is ambiguous in case features.

In conclusion, CG grammar writing required defining rules of linear morphosyntactic feature distribution based on linguistic theories, principles, and empirical observation. In the next section, I describe the processing of these CG rules in parsing texts.

4.2.4 Grammar Processing

In grammar processing, the CG rules are processed by a CG compiler (our case: VISL3)[42] for debugging and testing. For this purpose, I built a corpus. The corpus is a collection of Swiss German-language law texts with an automatically annotated morphological analysis (cf. Section 1.5). Because the CG-based tagger is a reductionist approach, it is important that every token in the corpus is augmented with morphological interpretations.

In the next subsection, I describe the creation of the corpus. I then present the evaluation of the rule-based tagger used on the test set deriving from the corpus.

4.2.4.1 Building a Corpus

In this section, I describe a corpus of Swiss German-language law texts enriched by morphological analysis, and I then select development data and test data.

Morphological Analysis

Swiss German-language law texts are augmented by morphological readings provided by the morphological analyzer Gertwol (Haapalainen and Majorin, 1994) and POS tagger (Schmid, 1999).

In the corpus, the completeness of morphological readings is of importance. Therefore, the primary issue is unknown words by Gertwol.[43] In the following subsection, I analyze the types of Gertwol's unknown words and describe the strategies for solving the problems (cf. Table 4.28).

[42] VISL3 is available at: `http://beta.visl.sdu.dk/` (last visited on 15/05/2013)

[43] Gertwol is a rule-based morphological analyzer and is implemented by a two-level finite state transducer. The two-level morphology analyzer consists of rules and lexicons and provides all possible morphological analyses of a German word. However, Gertwol does not provide any analysis if the stem is not in the lexicons. According to Haapalainen and Majorin (1994), the Collins German-English dictionary is used as the primary resource.

Type of unknown word problems	Strategies
Regional orthographic differences	String pattern matching and substitution
Lexical words	Pattern matching of affixes and the use of POS tags
Function words	Pattern matching of word forms
Abbreviations	String pattern matching

TABLE 4.28: Unknown word problems by Gertwol

Regional Orthographic Difference In the orthographic conventions of the German language, some differences exist between those written in Germany and in Switzerland. For example, unlike the standard German written in Germany, in Switzerland, the double "s" *ß* is not written as *ß* but as *ss*. Gertwol was designed to use standard German and follow the orthographic rules used in Germany. To adopt the German language as it is written in Switzerland, the words containing the string *ss* are capitalized because there is no capitalization of ß and thus the difference between *ß* and *ss* is absent in capitalized words. Hence, originally unknown tokens, such as *weiss* or *Strassenbahn*, are morphologically analyzed by Gertwol, by providing the capitalized forms, i.e. *WEISS, STRASSENBAHN*.[44]

Lexical Words To provide a morphological analysis for words that are unknown in Gertwol, POS-tags generated by a POS tagger, TreeTagger, are used. TreeTagger is a decision-tree based tagger and robust in unknown word problems (Volk and Schneider, 1998). However, TreeTagger has difficulties in assigning POS tags to morphologically ambiguous words, especially if they need knowledge about the contexts at wide distances. POS tags for verbs are thus one of the known weaknesses of TreeTagger (Volk and Schneider, 1998). To convert unknown words into cohorts, POS tags are changed to coarser POS tags. The POS tags of nouns, `NN` and `NE`, are regarded as one class of nouns. The POS tags for verbs or predicates (`VVFIN, VVINF, VVPP, and ADJD`) are transformed into two classes: affix *-t* or *-e(l)n*.[45] Depending on the class, POS tags are converted into corresponding cohorts. In addition, the first tokens of sentences with a verbal suffix are often erroneously analyzed as verbs. They are converted into cohorts of nouns and verbs. For example, *beiziehen* in the sentence (98) is unknown in Gertwol. TreeTagger analyzes the token *beiziehen* as a finite full verb, `VVFIN`. However, the correct

[44]The capitalization of words leads to generating undesired morphosyntactic categories, particularly those of nouns. As the rule-based supertagger is a reductionist approach, over-generated morphosyntactic analyses are less problematic than unknown morphosyntactic analyses are.

[45]These classes correspond to verbs in the third person singular and plural. Verbs in the first and second person hardly ever occur in law texts. First person and second person are used in narrative texts and colloquial speech.

analysis will be an infinite full verb, such as `VVINF`. Because *beiziehen* belongs to a verbal category with an affix *-e(l)n*, the corresponding morphological analyses are generated as shown in (99).[46]

(98) Dazu kann sie Sachverständige beiziehen.

(99) *beiziehen*
"beiziehen" V INF
"beiziehen" V IND PRÄS PL1
"beiziehen" V IND PRÄS PL3
"beiziehen" V KONJ PRÄS PL1
"beiziehen" V KONJ PRÄS PL3

Function Words Gertwol misses some morphological interpretations for function words. For example, Gertwol analyzes *zulasten* as a verb but not as a preposition. Missing morphological readings are added, individually based on word forms as illustrated in Table 4.29.[47]

Missing morphological analyses	**Examples**
Preposition	*zulasten, zuhanden, unbesehen, seitens, betreffend*
Verbal prefix	*bekannt, wahr, fest*
Pronominal adverb	*-einander* e.g. *miteinander, aufeinander*
Subordinating conjunction	*warum, wann, wo, wobei, wonach, wovon, wobei, wofür, womit, worauf, worunter, wodurch, inwieweit, weshalb*

Table 4.29: Missing morphological analyses by Gertwol

[46] V = verb, IND = indicative, INF = infinitive, PRÄS=presence, PL1=plural 1st person,PL3=plural 3rd person, KONJ = conjunctive

[47] Linguistically, wh-words (e.g. *warum, wo*) are adverbs or pronominal adverbs. As they behave like subordinating conjunctions and complementizers, they are additionally analyzed as subordinating conjunctions.

Abbreviations Gertwol does not handle abbreviations well. For example, BAFU, the acronym of *Bundesamt für Umwelt*, is typically unknown in Gertwol. Gertwol generates the morphological analysis of abbreviations (`ABK S`) for unknown words if they are matched to string patterns, such as all capital letters or alphanumeric letters.

Data Selection for Grammar Development and Test

The corpus is divided into a development set and a test set for grammar development and testing. The development set is used for the creation of CG rules and consists of 50 law documents that were selected from the entire body of Swiss German-language law texts, balanced in topics, from 2007 to 2011.[48]

The test set used for the evaluation contains 400 sentences (10,174 tokens). These sentences were selected from the corpus, excluding the development set, by using a stratified random sampling strategy. To sample the population of 400 sentences reflected in the distribution of sentences with regard to sentence length, the frequencies of subgroups of sentence length were counted in the whole collection of Swiss German-language law texts (cf. Table 4.30). The sentence length was determined by the number of tokens in a sentence. Based on the distribution, 10 stratums were constituted. The size of the samples in each stratum was proportional to that of the population. In each stratum, the samples were randomly selected. Hence, the test set was balanced in sentence length, and long sentences were included in the test set.

[48] The selection of this data set was carried out by Rebekka Bratschi. Swiss law texts are systematically classified into nine topics:
(1) State, people, and authorities
(2) Private law, administration of civil justice, and enforcement
(3) Criminal law, administration of criminal justice, and execution of sentences
(4) Education, science, and culture
(5) National defense
(6) Finance
(7) Public works, energy, and transport
(8) Health, employment, and social security
(9) Economy and technical cooperation

Sentence length (# of tokens)	**Population size (sentences)**	**Sample Size (sentences)**	**Percent distribution (%)**
5–14	47,543	116	29
15–24	55,425	136	34
25–34	30,629	76	19
35–44	13,988	32	8
45–54	6,699	16	4
55–64	3,445	8	2
65–74	2,026	4	1
75–84	1,265	4	1
85–94	952	4	1
95–	2,792	4	1
Total	164,764	400	100

TABLE 4.30: Test set: Population size, sample size and proportion of subgroups in sentence length

The minimal size of the test set required for investigating the performance of a tagger on Swiss German-language law texts was fulfilled by the 400 sentences. By assuming that the sample unit of language data is a sentence, and the population consists of Swiss German-language law texts (i.e. 165,188 sentences), the minimum size of the samples was 384 sentences if the confidence level was 95% (i.e. that the sample estimation was 95% certain) and the margin of error was ±5% (i.e. the sample estimation was in a range above and below the 5% of the estimation). The sample size is calculated using a normal distribution. At least 384 sentences were needed to test the targeted population at a reasonable certainty and error rate.

For the test, I manually annotated the test set according to the annotation schema of the topological dependency tag set (cf. Subsection 3.2.1). Prepositions were tagged as `PREP` and were not distinguished in `OBJP` or `PP`.

4.2.4.2 Evaluation and Discussion

In this section, I evaluate the CG-based tagger on the test set (400 sentences, 10,174 tokens) derived from the corpus presented in the previous subsection. The size of the tag set to be evaluated consisted of 38 (17 topological field tags and 21 dependency grammar tags).

This section is organized as follows. First, I discuss the evaluation methods used for tagging and then describe the baseline tagger and the reference parser. I then compare

the performance of our CG-based tagger with them. Finally, I present the results of the evaluation and provide an error analysis.

Evaluation Methods The following types of measurements are used in the evaluation: 1) label accuracy; 2) average tagging perplexity; and 3) precision, recall, and F1 score (Paroubek, 2007, van Halteren, 1999). Label accuracy is used in the CoNLL 2007 Shared Task on Dependency Parsing (Nivre et al., 2007) and the CoNLL 2007 Shared Task on Dependency Parsing (Buchholz and Marsi, 2006) to measure the percentage of correctly predicted dependency grammar relations (cf. Equation 4.1). Because the CG-based tagger assigns more than one tag per token, the number of tags labeled per token by the tagger is averaged (cf. average tagging perplexity; Equation 4.2). In addition, precision and recall are used to evaluate the tagger. The measurements of precision and recall are traditionally used in information retrieval. Precision is the ratio of the retrieved relevant documents to the number of documents returned by the system. Recall is the ratio of the retrieved relevant documents to the number of relevant documents. In tagging, precision is the ratio of the correct tags to the number of tags returned by the tagger (cf. Equation 4.3).[49] Recall is the ratio of the correct tags to the number of gold tags (cf. Equation 4.4). To harmonize precision and recall and to unify them to a measurement, F_1 score is computed (cf. Equation 4.5).

$$\text{Label accuracy} = \frac{\text{The number of correctly labeled tags}}{\text{The total number of tokens}} \tag{4.1}$$

$$\text{Average tagging perplexity} = \frac{\text{The total number of tags labeled by the tagger}}{\text{The total number of tokens}} \tag{4.2}$$

$$\text{Precision} = \frac{\text{The number of correctly predicted tags}}{\text{The number of tags assigned by the tagger}} \tag{4.3}$$

$$\text{Recall} = \frac{\text{The number of correctly predicted tags}}{\text{The number of tags present in the gold standard}} \tag{4.4}$$

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4.5}$$

[49] As the denominator is the total number of tags assigned by the tagger, the tagger is penalized in precision if it returns more than one tag per token.

Baseline Tagger To compare the rule-based supertagger with a statistical tagger, I use the baseline bigram Hidden Markov Model (HMM) tagger. HMM was selected from the available statistical tagging methods because it is the basic method used for statistical tagging in NLP. A bigram HMM tagger assigns the most probable tag, given the previous tag, combined with the probability of the current tag of the given word (cf. Appendix G.1).

To train the HMM tagger, 60% of the dependency version of TüBa-D/Z (cf. Section 1.5) are used.[50] The training set (`TüBa60Training` in Table 4.31) contains 700,882 tokens and 39,314 sentences. Because TüBa-D/Z is annotated with dependency grammar relations, but not with topological fields, topological field tags are automatically converted from the dependency grammar annotations in CoNLL (cf. Appendix D). The baseline tagger is then trained on the automatically converted TüBa data (`TüBa60TrainingTopo`). For the training, the implementation of hunpos[51] is used with the suffix-guessing of a maximum of four characters. In this way, morphologically rich German words are well handled in HMM.

Reference Parser For the evaluation, I use the state-of-the-art statistical graph-based parser developed by Bohnet (2010). I used the same topological dependency parser as used in Section 2.5.[52] The Bohnet parser is trained on the same training data (`TüBa60TrainingTopo`) as the Baseline HMM tagger.

Evaluation Results I estimate the overall performance of our CG-based tagger by using label accuracy and comparing it with that of the baseline tagger and the reference parser. Table 4.31 provides an overview of the results. The CG-based tagger achieved a label accuracy of 92.87% on the test set (cf. `Law400test` in Table 4.31).[53] In contrast, the baseline bigram HMM tagger achieved a label accuracy of 77.61% and the reference parser achieved a label accuracy of 88.96%. The CG-based tagger outperformed both the

[50] Ideally, the development sets are the same or derived from the same domain for the comparison of two systems. Because there is not a large amount of the in-domain data on which a HMM tagger is trained, an existing newspaper corpus TüBa-D/Z is used.

[51] Hunpos is available at: `https://code.google.com/p/hunpos/` (last visited: 23/03/2015)

[52] Bohnet parser is available at: `http://code.google.com/p/mate-tools/` (last visited on 01/06/2015). I used graph-based version anna-3.6 of the parser and trained the data with first, second, and third order factors.

[53] `Law400test` is the same data set used in Section 2.4. Because the CG-based tagger generates more than one tag per token, the label accuracy is computed as follows:

$$\text{Label accuracy} = \frac{\text{The number of correctly labeled tags}}{\text{The number of correctly labeled tags} + \text{The number of wrongly labeled tags}}$$

HMM tagger and the parser. In addition, the average tagging perplexity is estimated. The CG-based tagger generated 1.02 tags per token on average. Because the HMM tagger and the Bohnet parser assign one tag per token, the average tagging perplexity is 1 (cf. Table 4.31).[54]

	HMM tagger	Bohnet parser	CG Supertagger
Development set	TüBa60TrainTopo	TüBa60TrainTopo	-
Test set	Law400test	Law400test	Law400test
Label accuracy	77.61% (8017, 2313)	88.96% (9190, 1140)	**92.87**% (9846, 756)
Average tagging perplexity	1.00	1.00	1.02

TABLE 4.31: Overall label accuracy and average tagging perplexity of the HMM tagger, the statistical parser and our CG-based tagger:
label accuracy = % (# of correctly predicted tokens, # of incorrectly predicted tokens)

In addition, I estimate the performance of each topological dependency grammar label by using precision, recall, and the F_1 score. Table 4.32 and 4.33 provide an overview of the results. In topological field tagging, our CG-based tagger outperformed the HMM tagger and the statistical parser in all topological field tags (cf. Table 4.32). The supertagger achieved the high F1 score of above 0.8 in 11 of 12 topological field tags, whereas the HMM tagger achieved 4 of 12, and the statistical parser achieved 5 of 12. In addition, Table 4.32 shows that the results for left brackets were higher than those for right brackets, regardless of the system used. The left brackets are relevant cues for the recognition of the right brackets. In Chapter 2, I discussed the domain differences of the statistical parser (cf. Table 2.15). Compared with the performance of the statistical parser tested on the same text type as the training data (TüBa), that is, the upper bound of the statistical parser's performance (cf. Table 2.15), the CG-based tagger was better with the exception of two topological field categories. The results showed that the topological fields were handled well by our CG-based tagger. The supertagger is based on the linear distributions of morphosyntactic categories in wide contexts without probability accounts. The topological fields followed the rigid linear order of clause brackets, which were well recognized by the CG rule-based tagger.

[54] As discussed in Chapter 2, the domain difference affects the accuracy of automatic syntactic annotation: the Bohnet parser trained on TüBa-D/Z achieved a label accuracy of 90.89% on the test set of TüBa-D/Z. Because our supertagger is developed to achieve a high accuracy on law texts and is a domain-specific supertagger, I did not test it on TüBa-D/Z.

Tags	HMM tagger		Bohnet parser		CG supertagger	
LB-V1	0.67	(0.55, 0.85)	0.84	(0.79, 0.89)	**0.99**	(0.98, 1.00)
RB-V1	0.00	(0.00, 0.00)	0.55	(0.38, 1.00)	**0.87**	(0.87, 0.87)
LB-V2	0.88	(0.88, 0.88)	0.93	(0.97, 0.90)	**1.00**	(1.00, 1.00)
RB-V2	0.64	(0.72, 0.57)	0.83	(0.89, 0.78)	**0.93**	(0.94, 0.91)
LB-VF-CONJ	0.88	(0.92, 0.84)	0.91	(0.89, 0.93)	**0.98**	(0.99, 0.98)
RB-VF-CONJ	0.48	(0.46, 0.50)	0.64	(0.53, 0.81)	**0.88**	(0.85, 0.92)
RB-VF-RC	0.28	(0.20, 0.44)	0.68	(0.58, 0.80)	**0.94**	(0.92, 0.96)
LXCOMP	0.80	(0.67, 1.00)	0.55	(1.00, 0.38)	**1.00**	(1.00, 1.00)
-PUNCT-=LXCOMP	0.00	(0.00, 0.00)	0.00	(0.00, 0.00)	**0.72**	(0.75, 0.69)
RXCOMP	0.34	(0.65, 0.23)	0.61	(0.85, 0.47)	**0.83**	(0.86, 0.80)
TOP	0.00	(0.00, 0.00)	0.00	(0.00, 0.00)	**0.95**	(0.90, 1.00)
VC	0.82	(0.84, 0.80)	0.89	(0.86, 0.93)	**0.92**	(0.90, 0.94)

TABLE 4.32: Evaluation of topological field tagging: F_1 (precision, recall)

In dependency grammar tagging, the supertagger achieved a high F1 score of above 0.8 in 16 of 23 dependency grammar relations. The HMM tagger reached a F1 score of above 0.8 in 8 of 23 categories, and the statistical parser scored in 13 of 23 categories. Therefore, these results showed that the CG-based tagger outperformed both the HMM tagger and the statistical parser. In particular, the supertagger achieved a high performance in predicting the following dependency relations: the dependents of nouns (i.e. determiners, attributive adjectives), the head of nouns (i.e. comparative conjunction), and the modifiers of nouns (i.e. coordinating nouns). However, the supertagger did not outperform the other systems in identifying some syntactic functions of arguments, such as the accusative object (`OBJA`), the dative object (`OBJD`) of full nouns and pronouns, and the subjects (`SUBJ=LB-VF-RC`) and objects (`OBJA=LB-VF-RC`, `OBJD=LB-VF-RC`) of relative pronouns. In the grammatical functions of the arguments of full nouns and pronouns, the parser outperformed other systems. The HMM tagger performed poorly, which indicates that statistical methods are an advantage, but the previous and current tokens did not provide enough evidence for these dependency grammar relations. In the dependency grammar functions of the left bracket of a relative clause i.e. `LB-VF-RC`, the prediction was difficult for all systems because they are a mixture of topological fields and dependency grammar relations. The results of the HMM tagger indicated that the grammatical functions of relative clauses could be better predicted by using the probability estimation based on the previous tag and the current word form.

Tags	HMM tagger		Bohnet parser		CG supertagger	
SUBJ	0.47	(0.45, 0.49)	**0.84**	(0.82, 0.85)	**0.84**	(0.86, 0.82)
OBJA	0.31	(0.31, 0.30)	**0.76**	(0.80, 0.73)	0.69	(0.65, 0.74)
OBJD	0.17	(0.11, 0.45)	**0.62**	(0.50, 0.82)	0.55	(0.85, 0.41)
OBJG	0.00	(0.00, 0.00)	0.00	(0.00, 0.00)	**0.50**	(0.33, 1.00)
GMOD	0.43	(0.36, 0.54)	**0.94**	(0.94, 0.93)	0.88	(0.86, 0.89)
PN	0.63	(0.79, 0.53)	0.96	(0.94, 0.98)	**0.97**	(0.97, 0.98)
PRED	0.00	(0.00, 0.00)	0.28	(0.31, 0.25)	**0.64**	(0.57, 0.73)
ZEIT	0.00	(0.00, 0.00)	0.00	(0.00, 0.00)	**1.00**	(1.00, 1.00)
APP	0.00	(0.00, 0.00)	0.00	(0.00, 0.00)	**0.87**	(0.89, 0.85)
SUBJ=LB-VF-RC	**0.84**	(0.87, 0.82)	0.82	(0.72, 0.94)	0.66	(0.52, 0.91)
OBJA=LB-VF-RC	0.44	(0.29, 1.00)	**0.53**	(0.57, 0.50)	0.16	(0.58, 0.09)
OBJD=LB-VF-RC	**0.50**	(0.33, 1.00)	0.33	(0.33, 0.33)	0.31	(0.50, 0.22)
DET=LB-VF-RC	0.62	(0.80, 0.50)	0.60	(0.60, 0.60)	**0.82**	(0.70, 1.00)
PN=LB-VF-RC	0.60	(0.43, 1.00)	0.25	(0.14, 1.00)	**0.80**	(0.86, 0.75)
KON	0.84	(0.76, 0.93)	0.90	(0.86, 0.94)	**0.96**	(0.95, 0.98)
CJ	0.44	(0.32, 0.67)	0.89	(0.90, 0.89)	**0.92**	(0.87, 0.97)
DET	0.98	(0.99, 0.98)	0.98	(0.97, 0.98)	**1.00**	(1.00, 0.99)
ATTR	0.92	(0.93, 0.90)	0.93	(0.94, 0.91)	**0.97**	(0.96, 0.98)
PREP	**1.00**	(1.00, 1.00)	**1.00**	(1.00, 1.00)	0.98	(0.98, 0.98)
KOM	0.88	(0.80, 0.97)	**0.99**	(0.97, 1.00)	**0.99**	(1.00, 0.97)
ADV	0.86	(0.83, 0.90)	0.86	(0.83, 0.89)	**0.87**	(0.91, 0.83)
PART	0.00	(0.00, 0.00)	0.00	(0.00, 0.00)	**0.80**	(0.67, 1.00)
-PUNCT-	**0.99**	(1.00, 0.99)	0.97	(0.97, 0.98)	**0.99**	(0.99, 0.99)

TABLE 4.33: Evaluation of dependency grammar tagging:
F_1 (precision, recall)

Error Analysis A strength of a CG-based tagging is that it can integrate grammar constraints in wide context windows. To determine whether the performance of the CG-based tagger is affected by the length of sentences, the test set was divided into three subgroups: short sentences (5-14 tokens per sentence), medium-length sentences (15-24 tokens per sentence), and long sentences (25 tokens and more per sentence). They consisted of 116, 137, 147 sentences, respectively, and represent 29, 34, 37% of the test set, respectively. The CG-based tagger was tested on each subgroup. The results showed that all systems performed better in short sentences than long sentences, which was expected (cf. Table 4.34). In comparison with the statistical parser, the CG-based tagger outperformed the prediction of labels in long sentences, whereas the statistical parser performed slightly better than the CG-based tagger in short sentences.

Sentence length	**Size (%)**	**HMM tagger**	**Bohnet parser**	**CG supertagger**
Short	116 (29%)	79.11% (981, 259)	**95.28**% (1176, 64)	95.13% (1210, 60)
Medium	137 (34%)	77.48% (2023, 588)	**93.87**% (2437, 174)	**93.87**% (2526, 165)
Long	147 (37%)	77.39% (4940, 1443)	86.15% (5499, 884)	**92.00**% (6018, 523)

TABLE 4.34: Label accuracy dependent on sentence length: short sentences, medium-length sentences and long sentences

For the quantitative analysis of the error types of the tagger, I calculated the error rates of the pairs of an incorrectly predicted tag and its gold tag (cf. Table 4.35). The error rate is the ratio of the number of error pairs over the total number of errors. I divided the test set into four subsets, that is, 100, 200, 300, and 400 sentences in order to additionally investigate how many test sentences are required to find out the main error types. In each subset, I calculated the number of detected error types and the five worst predicted tag pairs. As illustrated in Figure 4.1, the number of types of errors newly found in the test set decreased with the increase in the size of the test set. This result indicated that an increase in the test size will gain a little discovery of error types for the effort of the annotation. In addition, the worst four types were ranked in the five worst tag pairs regardless of the test size (cf. (100)). This result indicated that a small test set was enough to discover the most frequent error types made by the tagger.

Test size	100 sentences	200 sentences	300 sentences	400 sentences
# of error types	86 error types	109 error types	129 error types	144 error types
Worst-1	SUBJ/OBJA (*11.1*)	SUBJ/OBJA(*11.3*)	SUBJ/OBJA (*11.0*)	SUBJ/OBJA (*11.2*)
Worst-2	OBJA/SUBJ (*8.4*)	OBJA/SUBJ(*9.3*)	OBJA=LB-VF-RC/SUBJ=LB-VF-RC (*9.2*)	OBJA=LB-VF-RC/SUBJ=LB-VF-RC (*9.0*)
Worst-3	OBJA=LB-VF-RC/SUBJ=LB-VF-RC (*8.0*)	OBJA=LB-VF-RC/SUBJ=LB-VF-RC (*8.7*)	OBJA/SUBJ(*8.9*)	OBJA/SUBJ(*8.6*)
Worst-4	OBJD/GMOD (*6.5*)	OBJD/GMOD (*6.2*)	OBJD/GMOD (*5.7*)	OBJD/GMOD(*5.7*)
Worst-5	RSK-V2/VC (*3.1*)	OBJD/OBJA (*3.8*)	OBJD/OBJA (*5.1*)	OBJD/OBJA (*5.2*)

TABLE 4.35: Evaluation of the CG-based tagger: predicted tag/correct tag (error rate (%))

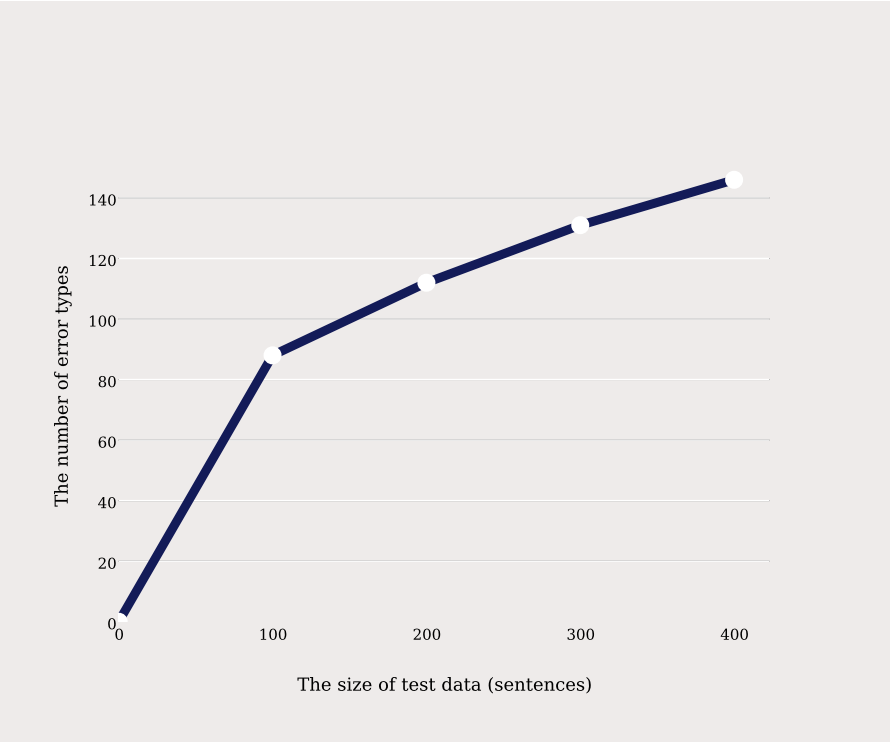


FIGURE 4.1: Number of error types and test set size

(100)
1. Subject (SUBJ) instead of accusative object (OBJA)
2. Accusative object (OBJA) instead of subject (SUBJ)
3. Accusative relative pronoun (OBJA=LB-VF-RC) instead of nominative relative pronoun (SUBJ=LB-VF-RC)
4. Dative object (OBJD) instead of genitive modification (GMOD).

The worst four error pairs (100) were dependency grammar relations. As expected, these pairs were ambiguous in case features because of case syncretism: nominative case vs. accusative case and dative case vs. genitive case. This result indicated that the supertagger, which was based on the hard linguistic constraints derived from the principles of linguistic theories, could not completely reduce ambiguities of case features, which led to multiple tagging.

To analyze the error types of the CG-based tagger thoroughly, I conducted an error analysis manually on 30 errors for each of five worst error pairs found in the test set (400 sentences). Table 4.36 shows the number of errors retrieved from the test set and the frequency of the error types found in the manual evaluation. The following paragraphs describe each error pair in detail.

Error pairs	Retrieved	Error types in analyzed 30 sentences
SUBJ/OBJA	85	Two ambiguous NPs (26), bare noun (3), coordination (1)
OBJA/SUBJ	68	Not ambiguous *die* (19), two ambiguous NPs:*die* (8), not ambiguous *das*
(LB-VF-RC)		(2), two ambiguous NPs:*welche* (1)
OBJA/SUBJ	65	Two ambiguous NPs (22), list (5), coordination (2), topological fields (1)
OBJD/GMOD	43	Ambiguous NP (30)
OBJD/OBJA	39	*sich* (21), bare noun (9)

TABLE 4.36: Frequent error types in the topological field tagging and dependency tagging in the test set

The error type, SUBJ/OBJA (predicted subject instead of accusative object), is typically caused by two ambiguous noun phrases. For example, the sentence (101) has two ambiguous noun phrases *Die Pferdepest-Zone* and *ein Gebiet*, which can be nominative or accusative. In the CG-based tagger, this case did not resolve the case ambiguity. The tagger assigned SUBJ and OBJA to *Gebiet* instead of only OBJA.

(101) Die Pferdepest-Zone umfasst ein **Gebiet** im Umkreis von ungefähr 100 km um die verseuchten Bestände.

The error type, OBJA/SUBJ of LB-VF-RC (predicted relative pronouns as accusative object instead of subject), is caused by the case ambiguity of the relative pronouns *die, das, welche*, which can be nominative or accusative. For instance, *die* in the sentence (102) is ambiguous in a clause because it is not the only candidate for arguments in a clause and *Daten* is also a candidate. In the CG-based tagger, the grammar rules are not sufficient to resolve these ambiguous relative pronouns.

(102) Behörden des Bundes und der Kantone, **die** Daten von sich in der Schweiz befindenden Asylsuchenden, anerkannten Flüchtlingen und Schutzbedürftigen an deren Heimat- oder Herkunftsstaat bekannt zu geben beabsichtigen, haben sich vorgängig beim BFM zu vergewissern, dass in erster Instanz das Asylgesuch abgelehnt oder ein Nichteintretensentscheid verfügt wurde oder dass durch die Bekanntgabe weder die betroffene Person noch deren Angehörige gefährdet werden.

The error type, OBJA/SUBJ (predicted accusative object instead of subject), is typically caused by two ambiguous noun phrases, such as the error pair SUBJ/OBJA. For example, in the sentence (101), the tagger assigned SUBJ and OBJA to *Pferdepest-Zone* instead of SUBJ because *Gebiet* is also ambiguous.

The error type, `OBJD/GMOD` (predicted dative object instead of genitive modifier), is typically caused by noun phrases that are ambiguous between dative and genitive and there is no dative object in a clause. For example, in the sentence (103), the tagger assigned `GMOD` and `OBJD` to *Projektarbeit* instead of only `GMOD`. In this sentence, there are no dative objects, so it is difficult to disambiguate noun phrases that are ambiguous between the dative and the genitive.

(103) Er umfasst überdies das angeleitete und betreute Verfassen oder Gestalten einer interdisziplinären **Projektarbeit**.

The error type, `OBJD/OBJA` (predicted dative object instead of accusative object), is typically caused by an ambiguous pronoun *sich* and bare nouns. In the sentence (104), for example, the tagger assigned `OBJD` and `OBJA` to *sich* instead of only `OBJA` because *sich* is ambiguous between the dative and the accusative, and there is no dative object in the clause. Because free dative objects are common, the removal of the dative case feature is difficult. In the sentence (105), the tagger assigns `OBJD` and `OBJA` to the bare noun *Anspruch*. *Anspruch* can be nominative, accusative, or dative. The nominative reading is discarded because of the mismatch in the subject-verb agreement.

(104) Soweit **sich** eine ausreichende Vorratshaltung an lebenswichtigen Gütern nicht durch Pflichtlager sicherstellen lässt (Art. 6-15 LVG), sichert es durch den Abschluss von Vereinbarungen mit Produktions-, Lagerhaltungs- und Dienstleis/-tungsbetrieben oder durch besondere Anordnungen die erforderliche Vorratshaltung.

(105) Ausländische Ehegatten und ledige Kinder unter 18 Jahren von Personen mit Niederlassungsbewilligung haben **Anspruch** auf Erteilung und Verlängerung der Aufenthaltsbewilligung, wenn sie mit diesen zusammenwohnen .

4.3 Conclusion

This chapter described the development of a rule-based topological dependency tagger. The evaluation showed that the CG-based supertagger achieved a label accuracy of 92.87% in the test set of Swiss German-language law texts. The system outperformed a bigram HMM tagger and a state-of-the-art statistical parser with label accuracies of 77.61% and 88.96%, respectively. These results indicated that a rule-based supertagger is a viable method for achieving high accuracy in domain-specific texts without the need for manually annotating the development data.

The CG-based supertagger presented in this chapter is based on hard constraints derived from linguistic theories and on psycholinguistic and domain-specific properties. The results showed that the rule-based supertagger was accurate in tagging topological fields but less accurate in tagging dependency grammar functions to arguments. Topological fields follow the rigid linear order of clause brackets, which were well recognized by the rule-based tagger. In contrast, grammatical functions do not follow the linear order of nouns. The hard linguistic constraints reduced the ambiguities of case features to some extent, but not completely. Where morphology failed to uniquely encode the case feature, the tagger generated multiple tags per token (1.03 tags per token on average). The incompleteness of disambiguation was mainly caused by case syncretism, that is, when the morphological suffixes of nouns and their dependents did not provide an unambiguous case feature. The results showed that the main problem of the rule-based tagger resided in the following case ambiguity pairs: nominative case vs. accusative case and dative case vs. genitive case. Furthermore, bare nouns caused ambiguous case features.

To select a dependency tag from multiple tags, the rule-based tagger must be extended by a component that can handle free word order (Lenerz, 1977). In the rule-based tagger, the canonical or default word orders of grammatical functions are not integrated because they can be freely scrambled. Scrambling is a linguistic phenomenon that is governed by multivariate factors, such as morphology, semantics, and pragmatics. In Chapter 5, a data-driven tagger is developed to model multivariate linguistic factors of word order scrambling.

Chapter 5

ML-Based Supertagger

Das Geschäftsreglement jedes Rates bezeichnet die zuständige Kommission.

– Federal Act on the Federal Assembly Art. 17 para. 1

Morphology and syntax are observable cues that are used to map words to morphosyntactic features and syntactic functions. However, morphology and syntax sometimes fail to encode the unambiguous case features and syntactic categories of arguments in German. The ambiguities are caused by morphological syncretism and free word order. The above sentence is a typical instance of this ambiguity: Morphologically, the words *Geschäftsreglement* and *Kommission* can be encoded as nominative or accusative, and the direct object precedes the subject.

The rule-based supertagger presented in Chapter 4 was not able to cope with this ambiguity. Hence, this chapter introduces a statistical method for overcoming this challenge. I present a statistical supertagger that assigns dependency grammar functions to nouns and prepositions. In statistical machine learning methods for tagging, tokens are labeled based on the linear order of the labels to be predicted and the features of current and nearby tokens. The use of a statistical tagging method is an advantage to probabilistically model the encoding of syntactic functions, particularly those belonging to the arguments in a clause. The statistical supertagger presented in this chapter takes into account the linguistic factors that interact in the context. The objective of developing this statistical supertagger is to achieve a high accuracy in analyzing Swiss German-language law texts.

This chapter is organized as follows. In the next section, I discuss the encoding of syntactic functions of arguments from a linguistic point of view. To model the encoding computationally, I then provide a review of statistical tagging methods. Based on this

review, I present my approach and discuss the modeling of syntactic encoding in a statistical tagging model. To optimize the tagging model for Swiss German-language law texts, I conduct experiments on the training data and the features. Finally, the newly developed statistical supertagger is combined with the rule-based supertagger (cf. Chapter 4) and state-of-the-art parsers in order to boost its accuracy.

5.1 Supertagging Problem: Free Word Order in German

This section provides a linguistic analysis to account for the encoding of syntactic functions of arguments (henceforth, grammatical functions) in German. In particular, I discuss the syntactic, semantic, and pragmatic features of the nouns that are integrated as linguistic factors into a statistical tagging model.

Canonical Word Order In German, there is canonical or unmarked word order as in other free word order languages. The list (1) provides an overview of the canonical word order of arguments in German. The canonical word order of subjects and objects in German follows a cross-linguistic tendency in terms of semantic role and animacy. A semantic role is a relation of arguments to the predicate in a clause. Arguments are categorized into agent, patient, recipient, and so on. Cross-linguistically, the agent is the most prominent semantic role, and it commonly precedes other semantic roles (Siewierska, 1993) (cf. (2)). The tendency of linearization reflects the 'actual development of events in the real world' (Siewierska, 1993). The natural process of an action flows from agent to patient and from agent to receiver (ibd.). Speakers tend to describe an event in terms of their experience the world (ibd.). The prominence of the agent further triggers the mapping to the most prominent syntactic function, which is the subject (cf. accessibility hierarchy (3)). In linguistic typology, two scales of prominence are aligned by means of harmony alignment (Prince and Smolensky, 2004, Aissen, 2002). In harmony alignment, the prominent category in a class is aligned with the prominent category in another class. The harmony alignment of (2) and (3) states that the agent is more likely to be mapped to the subject, and the patient is the preferred object (cf. (4)). The second account, animacy, is the 'agentness' of entities. Animacy distinguishes entities between persons, animals, and things. In the hierarchy of person (5), higher ranked entities precede lower ranked entities. Accordingly, a person is more likely to precede an animal and a thing. In particular, the first person has a strong tendency to be mentioned first in a sentence. Thus, hierarchy reflects the 'perception of salience' or 'empathy' (in Kuno's term (Kuno and Kaburaki, 1977)) that is, experience in the world (Siewierska, 1993). Humans cannot feel more empathy than they do about themselves (ibd.). The

linearization of the appearance of entities reflects the speaker's point of view. Because animate entities tend to precede inanimate objects and are mapped to the subject (cf. harmonic alignment (6-c)), a subject aligned to a human is more likely to precede an object denoting an inanimate entity.

(1) The preference of the linearization of arguments:[1]

a. Canonical word order (pronouns): SUBJ > OBJA > OBJD

b. Canonical word order (full nouns): SUBJ > OBJD/OBJA

(2) The semantic role hierarchy and precedence
Agent > patient/recipient > benefactive > instrumental > spatial > temporal

(3) Keenan-Comrie accessibility hierarchy (Keenan and Comrie, 1977)[2]
SUBJ > OBJA|OBJD.

(4) Harmonic alignment of syntactic functions of arguments and semantic roles:[3]
Agent/SUBJ >> Agent/OBJA|OBJD
Patient|Recipient/OBJA|OBJD >> Patient|Recipient/SUBJ

(5) The person hierarchy and precedence (Siewierska, 1993)
1st person > 2nd person > 3rd person human > higher animals > other organism > inorganic matter > abstracts

(6) a. Hierarchy of animacy (simplified from (5))
human > animal > thing

b. Hierarchy of syntactic functions of arguments (the same as (3))
SUBJ > OBJA|OBJD

c. Harmonic alignment:
SUBJ/human >> SUBJ/animal >> SUBJ/thing
OBJA|OBJD/thing >> OBJA|OBJD /animal >> OBJA|OBJD/human

[1] The linear preference of arguments is marked with >.

[2] Originally, Keenan and Comrie (1977) proposed the following hierarchy: SUBJ(SU) > OBJA(DO) > OBJD (IO) > PN (OBL) > GMOD (GEN) > CJ (OCOMP). As the order of patient (OBJA) and recipient (OBJD) is cross-linguistically controversial (Siewierska, 1993), these two objects are regarded as one class as OBJA|OBJD.

[3] >> means: "stronger than"

In German, the canonical order of dative and accusative objects in a ditransitive sentence varies depending on the 'pronominality' of nouns. In pronouns, direct objects tend to precede indirect objects. In full nouns, the linearization depends on the animacy of objects (Steinbach and Vogel, 1998). Ditransitive verbs, such as *geben, schenken, gönnen*, require an animate entity as indirect object and an inanimate entity as a direct object (cf. Example (7)), whereas other ditransitive verbs, such as *entziehen, ausliefern, aussetzen*, require an inanimate entity as the indirect object and an animate entity as the direct object (cf. Example (8)).[4] Cross-linguistically, persons tend to precede things (cf. (5)). Thus, the canonical word order of a ditransitive construction in German follows the linearization of animacy in linguistic typology.

(7) Müller (1999)

a. [Canonical] daß man dem Fritz das Buch gegeben/geschickt hat

b. ?daß man das Buch dem Fritz gegeben/geschickt hat

(8) Müller (1999)

a. [Canonical] daß man die Kinder diesem Einfluss entzogen/ausgeliefert hat

b. ?daß man diesem Einfluss die Kinder entzogen/ausgeliefert hat

Furthermore, pronouns tend to be placed before full nouns in the mittelfeld. The precedence of pronouns is referred to as the Wackelnagel position (Sternefeld, 2006, pp. 352). For example, the pronouns *diese ihn* in the sentence (10) precede the full noun *Auffangeinrichtung*. However, the Wackelnagel position is not obligatory, but a mere tendency[5] as it competes with the canonical word order. While the sentence (10) follows the Wackelnagel position, the canonical word order - subject precedes object - outranks it in sentence (11).

[4]Sentences (7) and (8) showed that canonical word order, as determined by animacy, could not be freely scrambled. Müller claimed that D-structure is OBJA-OBJD, not OBJD-OBJA. His analysis of scrambling underlies a different assumption of the input of scrambling. However, his examples showed clearly that animacy triggers canonical word order and restricts scrambling.

[5] Pronominal direct objects and indirect objects can be placed after full nouns in the mittelfeld as shown in the examples (i) and (ii), respectively (cf. Sternefeld, 2006, pp. 353). According to Müller (1999), the Wackelnagel position is obligatory if the pronouns are not phonologically stressed.

(i) Ein Aufzug gilt als übertragen, sobald der Montagebetrieb *ihn* dem Benutzer erstmals zur Verfügung stellt.

(ii) Der Vorstand hat das Recht und die Pflicht, nach den Befugnissen, die die Statuten *ihm* einräumen, die Angelegenheiten des Vereins zu besorgen und den Verein zu vertreten.

(9) Canonical word order (pronominality):
Pronoun > Full noun

(10) Kommt der Arbeitgeber der Aufforderung der Ausgleichskasse der AHV nicht fristgemäss nach, so meldet *diese ihn* der Auffangeinrichtung (Art. 60) rückwirkend zum Anschluss.[6]

(11) Vom Beginn der Abenddämmerung an bis zur Tageshelle und wenn die Witterung *es* erfordert, müssen die Fahrzeuge beleuchtet sein.

In summary, canonical word order in German follows a cross-linguistic tendency in terms of semantics. The pronominality factor also affects word order in German.

Word Order Freezing - Condition on Scrambling However, in German, word order is relatively freely scrambled. Thus, direct objects can be placed before subjects (e.g. (12)), indirect objects can precede subjects (e.g. (13)), and direct objects can be followed by indirect objects (e.g. (14)). However, in some cases, canonical word order cannot be scrambled. Canonical word order must be preserved (or frozen) if it is the only strategy used to encode grammatical functions. Word order freezing has been reported in German (Vogel, 2004) as well as in other free word order languages, such as Dutch (Zeevat, 2006), Japanese (Flack, 2007), Korean (Lee, 2001), and Hindi (Lee, 2001).

(12) *Den Umrechnungskurs* bestimmen *die Vertretungen* nach Weisung des Eidgenössischen Departements für auswärtige Angelegenheiten.

(13) Für den einem Dritten zugefügten Schaden haftet *dem Geschädigten die Organisation* nach den Artikeln 3-6.

(14) Wer durch eine Handlung, einen Entscheid oder eine Unterlassung derselben in seinen Interessen verletzt wird, kann *diesen Sachverhalt der FINMA* anzeigen.

In German, word order freezing is triggered by morphology and world knowledge (Vogel, 2004). The sentence pair (15) is a typical example in which canonical word order (15-a) cannot be scrambled. The scrambled word order (15-b) is not interpreted as *Hans* loves

[6] All examples in this chapter are derived from Swiss German-language law texts (cf. Section 1.5) if they are not specially marked.

Maria.[7] *Hans* and *Maria* are both morphologically ambiguous and can be nominative or accusative. Therefore, theoretically, they can be the subject or direct object of the predicate *love*. Semantically, *Maria* and *Hans* are both persons. In written language, the only means for the encoding of arguments is canonical word order *subject-object*, which affects the interpretation of the sentence (15-a).

(15) 'Hans loves Maria' (Hans = SUBJ, Maria = OBJA)

a. Hans liebt Maria. (Vogel, 2004)

b. #Maria liebt Hans.[8] (self-created)

However, canonical word order is not frozen, that is, it can be scrambled if the morphological markings or semantics are not ambiguous to encode grammatical functions. For example, the sentence (16-a) can be easily scrambled, such as in (16-b), because the morphology encodes grammatical functions. In the sentence (17-b), animacy encodes grammatical functions because an animate entity (here: *Maria*) is more likely to be the subject, whereas an inanimate entity (here: *Buch*) is more likely to be the object.

(16) 'Maria loves the dog' (Maria = `SUBJ`, Hund = `OBJA`)

a. Maria_`NOM|ACC|DAT` liebt den Hund_`ACC`. (self-created)

b. Den Hund liebt Maria. (self-created)

(17) 'Maria loves the book' (Maria = `SUBJ`, das Buch = `OBJA`)

a. Maria_`NOM|ACC|DAT` liebt das Buch_`NOM|ACC`. (self-created)

b. Das Buch liebt Maria. (self-rephrased)

The discussion so far has suggested that the encoding of grammatical functions is based on morphological marking, canonical word order, and animacy. However, in some cases of real data from law texts, these three linguistic cues do not encode grammatical functions. For example, in the sentence (18), *Nähere* and *Gesetz* are both morphologically ambiguous between the nominative and the accusative, and both are inanimate. However, the canonical word order *subject-object* does not encode the syntactic functions of arguments. In the legislative domain, specific laws such as civil code or a specific

[7] It is worth noticing that the sentence (15-b) can be interpreted as such with marked prosodic structures.

[8] This sentence is marked with a number sign #, as it cannot be interpreted that Hans is the one who loves Maria, unless it is prosodically marked.

ordinance act as agents. In this case, some world knowledge or domain knowledge is necessary to encode grammatical functions.

(18) Er erteilt ferner das Kantonsbürgerrecht, sofern die Erwerbung oder Zuteilung eines Gemeindebürgerrechts bereits erfolgt ist. Das *Nähere* bestimmt das *Gesetz*.

Free Word Order - Triggers for Scrambling Cross-linguistically, word order is affected by several common linguistic factors, such as syntactic, semantic, thematic, and pragmatic features (Siewierska, 1993). In this section, I discuss the word order scrambling in German that is triggered by information structure.[9]

Information structure is pragmatic structure in which thematic roles, such as topic ('aboutness'), completive focus ('new information'), and contrastive focus ('contrast') are marked by the linguistic means of word order, prosody, and adverbs such as *even* or *only*. In the realization of information structure, old information tends to precede new information, and prominent information usually precedes less prominent information (cf. (19)). Typically, the given entity in the discourse is associated with the topic, while the new entity is usually the focus. Focus is further categorized into completive and contrastive focuses (cf. (20)). While completive focus marks 'a new piece of information', contrastive focus designates an emphasis evoked by the alternatives (cf. Rooth, 1992) and is more prominent (cf. (20)). According to Choi (1996), new or not-prominent entities are not scrambled (i.e. placed forwards), whereas old or prominent entities undergo scrambling.

[9] Linearization is also affected by the heaviness of a syntactic unit. Cross-linguistically, a short and less complex syntactic unit is preferred to precede a long and complex syntactic unit (Siewierska, 1993) (cf. heavy NP shift in English (Arnold et al., 2000)).

(i) Formal hierarchy(Siewierska, 1993)
short/structurally simple > long/structurally complex

German is no exception. According to Behaghel's law ('Gesetz der wachsenden Glieder') (Behaghel, 1932, pp. 241ff), constituents can be placed discontinuously and separated into modifying and modified elements if longer elements precede shorter ones. For example, the sentence (ii) has the constituent *das Glück meiner Kindertage im lieben alten Wien* separated by the verb *schildern*. Behaghel's law is applicable to modern law texts: For example, according to this law, an infinitive clause is extraposed after the verb prefix *zu* in the sentence (iii). In addition, according to the Behaghel's first law 'the elements that belong together conceptually are placed close together' (Behaghel, 1932, pp. 4), the sentence has a marked word order. To be close to the modifying infinitive clause, the noun phrase *das Recht* is placed at the end of the clause, although it breaks the canonical word order.

(ii) *wie soll ich das Glück schildern meiner Kindertage im lieben alten Wien* (Behaghel, 1932, pp. 243)

(iii) Bei der Versicherung zugunsten Dritter steht dem Versicherer das Recht zu, die Prämienforderung mit der dem Begünstigten geschuldeten Leistung zu verrechnen.

(19) Linearization of information structure (Choi, 1996)

a. Discourse old >> Discourse new: *Discourse old entities precede discourse new ones*

b. Discourse prominent >> Discourse not-prominent: *Discourse prominent entities precede discourse not-prominent ones*

(20) Information structure (Choi, 1996)

a. Topic is discourse old and prominent

b. Tail is discourse old and not prominent

c. Contrastive Focus is discourse new and prominent

d. Completive focus is discourse new and not prominent

For example, the entity *Grosser Rat* in the last sentence of (21) was previously mentioned and therefore is old in the discourse. The entity *Grosser Rat* is the topic in the last sentence. Therefore, this direct object can be scrambled to realize the linearization of information structure, which leads to a violation of the canonical word order. In the last sentence in (22), the entity *Umrechnungskurs* is discourse new but was previously evoked by the entity *Landeswährung*. Therefore, it is a contrastive focus. The contrastive focus is scrambled; thus, the direct object precedes the subject *Vertretungen*. This also is a violation of the canonical word order because of the realization of information structure.

(21) Durch Gesetz können diesen Kommissionen bestimmte Entscheidungsbefugnisse aus den Zuständigkeiten *des Grossen Rates* übertragen werden. *Dem Grossen Rat* muss jedoch die Möglichkeit gewahrt bleiben, ein einzelnes Geschäft an sich zu ziehen.

(22) Im Ausland sind die Gebühren und Auslagen in der entsprechenden Landeswährung zu bezahlen. Das EDA kann abweichende Bestimmungen erlassen. *Den Umrechnungskurs* bestimmen die Vertretungen nach Weisung des EDA.

The preference for thematic linearization (19) can be explained by the familiarity principle and definiteness. Definite entities are often old and familiar, and indefinite entities are often new and unfamiliar. Cross-linguistically, familiar entities tend to precede unfamiliar entities (Siewierska, 1993) (cf. (23)). In German, definite and specific nouns precede indefinite and unspecific nouns (Müller, 2002) (cf. (24)). For example, the definite entity *Wein* in the sentence (25-a) is scrambled and precedes the indefinite entity

Kunde although the scrambling causes a permutation of the canonical word order *indirect object - direct object*. The definiteness of nouns also triggers a marked word order in law texts (cf. (26)).

(23) Familiarity linearization: Familiar > unfamiliar

(24) Definiteness linearization:
Definite nouns > Indefinite nouns

(25) Müller (1999)[10]

a. daß der Verkäufer den Wein einem Kunden empfahl

b. ?daß der Verkäufer einem Kunden den Wein empfahl.[11]

(26) Ein Ausländer kann jedoch durch letztwillige Verfügung oder Erbvertrag den Nachlass einem seiner Heimatrechte unterstellen.

In summary, canonical word order is a linguistic means to encode grammatical functions in German. Free word order in German is triggered by linguistic factors, such as syntax (e.g. pronominality), semantics (e.g. animacy) and pragmatics (e.g. information structure, definiteness), and world knowledge (e.g. domain knowledge). In the next section, to model the encoding of grammatical functions computationally and to incorporate the linguistic factors into a model, I describe the technical background of state-of-the-art machine learning (ML) methods used in tagging (or sequences).

5.2 ML-Based Tagging Methods

In NLP, machine-learning methods for sequences are mainly employed in POS tagging. In the history of POS tagging, data-driven POS tagging has improved significantly since 2002 through the work of Collins (2002) (cf. Manning, 2011, Spoustová et al., 2009). Collins (2002) proposed a feature-rich perceptron algorithm for tagging problems. The algorithm is simple, easy to implement, and fast. The structured perceptron algorithm and its variants have been widely applied in NLP, such as in POS tagging in English

[10] Müller (1999) claims that D-structure is OBJA-OBJD. In his work, the example is used as evidence that indefinite nouns cannot scramble.

[11] The question mark means that the sentence is marked and sounds odd.

and Czech (Spoustová et al., 2009), in Icelandic (Loftsson and Östling, 2013), in Chinese (Zhang and Clark, 2011). This algorithm has been applied for supertagging in HPSG (Zhang et al., 2012, 2009) and CCG (Clark and Curran, 2007), word segmentation (Zhang and Clark, 2011, 2007), term recognition (Jiampojamarn et al., 2006), morphological disambiguation (Sak et al., 2007), named entity recognition (Ciaramita and Altun, 2006, Carreras et al., 2003), word alignment (Moore et al., 2006, Moore, 2005), syntactic and semantic parsing (Lluís and Màrquez, 2008), query spelling correction (Li et al., 2012), and parsing (Li and Zhou, 2012, Zhang and Clark, 2011, Zhang and Nivre, 2011, Carreras et al., 2006, Collins and Roark, 2004). Furthermore, the structured perceptron is currently used in one of the best English parsers (Carreras et al., 2008). Recently, conditional random fields (Lafferty et al., 2001) have become state-of-the-art in sequential models. They have been widely used in NLP, e.g. in POS tagging in German (Sennrich et al., 2013), Bengali (Ekbal et al., 2007), Gujarati (Patel and Gali, 2008), Amharic (Adafre, 2005), English, Finnish, Czech, Estonian, and Romanian (Silfverberg et al., 2014). They have also been used in noun chunking (Nguyen et al., 2009, Sha and Pereira, 2003), word segmentation (Peng et al., 2004), supertagging in HPSG (Blunsom and Baldwin, 2006), parsing (Finkel et al., 2008), named entity recognition (Liu et al., 2011, Finkel et al., 2005, Settles, 2004, McCallum and Li, 2003), and in information extraction (Peng and McCallum, 2004).

The supervised sequence learning problem that we wish to approximate is the function $f : x_{1:n} \mapsto y_{1:n}$, where $x_{1:n}$ is an input sequence $x_1 \dots x_n$ and $y_{1:n}$ is an output sequence $y_1 \dots y_n$. In NLP, a sentence, such as *Meine Katze miaut*, is an input sequence that contains n words, such as *Meine, Katze, miaut.* For example, the output sequence is a sequence of topological dependency grammar relations, that is, `DET, SUBJ, LB-V2`. In supervised ML methods, the mapping function is learned from annotated training data, that is, known mapping pairs $(x_{1:n}, y_{1:n})$. However, the fundamental goal of the learning is beyond the already-seen training data, which is to generalize the prediction of the class sequence $y_{1:n}$ on unseen $x_{1:n}$.

In supervised probabilistic machine learning methods, there are two different paradigms: generative and discriminative learning. Generative learning methods learn the distribution of a joint probability $P(x_{1:n}, y_{1:n})$ in which both input $x_{1:n}$ and output $y_{1:n}$ are true. The hidden Markov model (HMM) is an example of a generative model. In discriminative learning methods, the posterior probability $P(y_{1:n}|x_{1:n})$ is directly learned. In posterior probability, $y_{1:n}$ is true if $x_{1:n}$ is true. The maximum entropy Markov model (MEMM) and the conditional random fields (CRF) model are examples of probabilistic discriminative models. In non-probabilistic machine learning, there is a hyperplane classifier, such as the perceptron and the support vector machine. The structured perceptron is a hyperplane sequential model.

To assign dependency grammar functions to nouns and prepositions, I use two feature-rich sequential ML methods: the CRF and the structured perceptron. A feature-rich learning algorithm makes use of linguistic cues in contexts. It makes the assumption that each tth input token x_t has linguistic properties such as word forms or POS tags, which provide a piece of information for the prediction of tags. The feature function here is a binary indicator function. Each dth feature is mapped to a binary score $\{0, 1\}$ to indicate the existence $(= 1)$ or non-existence $(= 0)$ of the linguistic property in the current token. Feature-rich ML methods have the great advantage that the feature function integrates rich linguistic information into models, compared to a standard HMM in which features are based solely on word forms. The MEMM is also a feature-rich ML method. However, MEMM suffers from a label bias problem because of the local normalization of the probability estimation (cf. Appendix G). Compared to the MEMM, the CRF is advantageous in dealing with long sequences because of the global normalization of the probability estimation. Since law texts tend to have long sentences, the choice of these methods is justified. Moreover, the structured perceptron and the CRF have not yet been applied to German in the context of supertagging.

The following subsections briefly outline these two feature-rich machine learning methods used in this thesis - the CRF model and the structured perceptron. The technical details of these tagging methods (HMM, MEMM, CRF and structured perceptron) are described in Appendix G.

5.2.1 Structured Perceptron

The structured perceptron algorithm proposed by Collins (2002) is a feature-rich and margin-based supervised learning algorithm for sequence models. The algorithm consists of a function $GEN(x)$, local feature function, global feature function and parameter updating.

A function $GEN(x)$ maps the input token x_t to the possible candidate tag set z_t. For each tth input token x_t paired with each ith candidate output tag z_t^i, the (local) feature functions compute a binary value $\{0, 1\}$ and build a feature vector. The feature vector is then mapped to a real-valued parameter (i.e. weight) vector w. By linearly combining these vectors, each input-output pair $(x_t^i,\ z_t^i)$ gets a score:

$$score(x_t^i, z_t^i) = \sum_{d=1}^{D} w_d f_d(x_t^i, z_t^i)$$

After having scored all input-output pairs $(x_{1:n}, z_{1:n})$ in a sequence, the best output tag sequence $z_{1:n}$ is approximated in a global feature function. To find the best sequence $z_{1:n}$, the Viterbi algorithm is used in Collins (2002) and beam search in Zhang and Clark (2011).

The learning algorithm of perceptron is online and error-driven. If the current parameter model predicts a wrong tag, then the parameters are immediately updated. Otherwise, the parameters are not changed. More specifically, if the predicted tag z_t is not equal to the gold standard tag y_t, the parameters are updated. Since the features are binary indicator functions, the updating is computed simply by adding 1 for the parameters of features which would lead to the gold standard tag y_t and by subtracting 1 for the parameters of features which were used for the wrong prediction z_t. The parameter updating is repeated for jth learning epoch. As Nguyen and Guo (2007) showed, the perceptron algorithm reaches the maximal performance after less than 10 iterations.

In averaged perceptron (Freund and Schapire, 1999, Collins, 2002), the parameters of the features are averaged in a batch at the end of the online parameter updating. The averaged weight vectors are calculated by dividing the sum of all weight parameter vectors of all learning passes by the total number of the learning passes. Hence, the parameters of the model are normalized regarding the number of times the parameters retain the values of the current model during online learning (Daumé III, 2014). As Collins (2002) shows, the averaged version of perceptron has outperformed the non-averaged standard version of perceptron.

5.2.2 Conditional Random Fields (CRF)

In this subsection, we outline the algorithm of CRF (Lafferty et al., 2001, Sutton and McCallum, 2011). CRF is a random field (also known as undirected graph or Markov network) for conditional probability $P(y_{1:n}|x_{1:n})$. To calculate the conditional probability, CRF makes use of the maximum entropy model and normalizes the probability globally in a sequence:

$$P(y_{1:n}|x_{1:n}) = \frac{1}{Z(x_{1:n})} \exp\left(\sum_{n=1}^{N}\sum_{d=1}^{D} w_d f_d(x_{1:n}, y_n, y_{n-1}, n)\right)$$

The term $Z(x_{1:n})$ sums over all possible values of the sequence $y_{1:n}$:

$$Z(x_{1:n}) = \sum_{y_{1:n}} \exp\left(\sum_{n=1}^{N}\sum_{d=1}^{D} w_d f_d(x_{1:n}, y_n, y_{n-1}, n)\right)$$

The parameter estimation is the maximization of the log conditional probability:

$$\arg\max_{w} \sum_{t=1}^{T} \log P(y_{1:n}^{t} | x_{1:n}^{t}) - \lambda \sum_{m=1}^{M} w_m^2$$

The regularization term *lamda* is $1/2\sigma^2$ which determines the magnitude of penalty for over-fitting. Thus, the goal of the learning is to find the optimal weight w which maximizes the regularized conditional log-likelihood.

In summary, I provided an overview of the statistical tagging methods, structured perceptron and CRF. CRF and perceptron are different in the calculation of parameters. However, the scoring of features $\sum_{d=1}^{D} w_d f_d(x_t^i, z_t^i)$ is identical and the same features are thus applicable to these ML methods.

In the next section, I address the question of how to incorporate the linguistic factors presented in the previous section into a sequential model that automatically assigns dependency grammar functions to nouns and prepositions.

5.3 ML-Based Supertagger

In this section, I develop a statistical supertagger that assigns 17 dependency grammar functions to nouns and prepositions (cf. Table 5.1).

Tag	Description	Examples
SUBJ	Subject	*Das **Gesetz** gilt für Wirbeltiere.*
OBJA	Direct object	*Der Bundesrat erlässt die **Ausnahmebestimmungen**.*
OBJD	Indirect object	*Die kantonalen Behörden eröffnen ihre Entscheide sofort der zuständigen **Bundesbehörde**.*
OBJG	Genitive object	*Der gewerbsmässige Handel mit Tieren und das Verwenden lebender Tiere zur Werbung bedürfen einer **Bewilligung**.*
GMOD	Genitive modification	*Der Bundesrat regelt die Einzelheiten der **Meldung**.*
APP	Apposition	*Handlungen nach Artikel 11 **Absatz** 1 letzter Satz sind verfahrensmässig Tierversuchen gleichgestellt.*
PN	Complement of preposition	*Globale Finanzhilfen werden in **Tranchen** ausbezahlt*
CJ	Complement of conjunction	*Niemand darf ungerechtfertigt einem Tier Schmerzen, Leiden oder **Schäden** zufügen, [...]*
KON	Complement of conjunction (comma)	*Niemand darf ungerechtfertigt einem Tier Schmerzen, **Leiden** oder Schäden zufgen, [...]*
GRAD	Grade	*Die Bodenfreiheit zwischen der Liegefläche und dem Trennbügel muss für Rinder mit mehr als 400 **kg** Körpergewicht mindestens 40 cm betragen.*
ZEIT	Time	*Der Kantonsrat wählt jedes **Jahr** aus der Mitte des Regierungsrates den Landammann und den Landstatthalter*
EXPL	Expletive	***Es** gelten die Bestimmungen des BWIS*
PRED	Predicate	*Milch- und Rahmanaloge sind **Öl-in-Wasser-Emulsionen** aus Wasser, Speiseölen und/oder Speisefetten mit weiteren Zutaten wie Milch, Milchprodukten, stärkehaltigen Erzeugnissen (z. B. Mehl ‚Stärke aus Getreide oder Kartoffeln) und Zuckerarten*
PAR	Parenthesis	*Alkoholabhängige Frauen, so **Winkler**, seien nicht nur aufgrund ihrer Krankheit stigmatisiert, [...]*
ROOT	Root	***Köpfchen**, Köpfchen!*
PP	Preposition (adjunct)	*Der Bundesrat bestimmt den Rahmen **für** die kantonalen Gebühren*
OBJP	Prepositional object	*Er sorgt **für** die Information der Bevölkerung über Tierschutzfragen.*

TABLE 5.1: Dependency labels for the ML-based tagger with examples

This section is organized as follows: First, I define the linguistic unit of a sequence and a set of features in order to model word order phenomena in German as described in Section 5.1. I then test the two sequential ML methods that are considered the most suitable for the task based on the review of various approaches in the previous section, that is, CRF and the structured perceptron. Based on the results of the experiments with the training data and feature sets, the model optimized for Swiss German-language law texts will be developed.

5.3.1 Approach: Application of Feature-Rich Sequential ML Models for Free Word Order in German

An important aspect in the design of a sequential model is to consider the kind of information that is integrated into prediction models. In concrete terms, we need to define the notions of a sequence and a set of features. Defining a sequence requires determining the kinds of tokens that are near in the context and which ones have to be learned as a unit. In defining features, we have to examine the kinds of information in the near context that are useful for the prediction. Because the near context is defined as the notion of a sequence, the next section defines the sequence before defining the features.

5.3.1.1 Sequences

A sequence is a unit in which a sequential model learns the parameters and decodes the output based on a given sequence of input tokens (cf. Section 5.2). To model the linearization of arguments in German (cf. Section 5.1) and head-modifier relations (27) directly, I define a sequence as (28).

(27) Linear order of modifiers: (>> means precedence)

a. Full noun >> `GMOD`: *Full nouns precede genitive modifiers (`GMOD`)*

b. Preposition >> `PN`: *Prepositions precede modifiers of prepositions (`PN`)*

c. `KON` >> `CJ`: *Coordinated tokens with commas (`KON`) precede the coordinated token with coordinating conjunction (`CL`).*

(28) a. A sequence contains only nouns, pronouns and prepositions.

b. A sequence corresponds to a clause.

According to the definition of (28-a), nouns, pronouns, and prepositions interact directly within a sequence. The linear patterns of tokens in the near context are directly coded as features. For example, we consider sentences (29). By removing all tokens except nouns, pronouns, and prepositions, the input sequences contain only the tokens in which we are interested (cf. (30)). The sequences of dependency grammar labels (31) are the targeted output sequences. Hence, the prediction of the dependency grammar label for *Sie* in the sentence (29-c), for example, is conditioned by the property of the token itself and the tokens interacting around it, e.g. *Rundschreiben, Mitteilungen* in (30-c). To automatically extract nouns, pronouns and prepositions, I use the rule-based supertagger (cf. Chapter 5).

(29) Raw data:

a. Dem Entscheidverfahren geht ein Schlichtungsversuch vor einer Schlichtungsbehörde voraus.

b. Der ETH-Rat ernennt auf Antrag der Präsidentin oder des Präsidenten der ETH die Professorinnen und Professoren.

c. Sie erlässt Rundschreiben, Mitteilungen und Stellungnahmen von allgemeiner Tragweite.

(30) Input data for the structured perceptron and CRF:

a. Entscheidverfahren, Schlichtungsversuch, vor, Schlichtungsbehörde

b. ETH-Rat, auf, Antrag, Präsidentin, Präsidenten, ETH, Professorinnen, Professoren

c. Sie, Rundschreiben, Mitteilungen, Stellungnahmen, von, Tragweite

(31) Gold labels (in training) and output labels (in prediction):

a. `OBJD SUBJ PP PN`

b. `SUBJ PP PN GMOD CJ GMOD OBJA CJ`

c. `SUBJ OBJA KON CJ PP PN`

In addition, a sequence is defined as a clause instead of a sentence (cf. (28-b)). A clause is a linguistic unit in which the arguments of a verb interact uniquely with each other function[12] and exhibit linguistic patterns (cf. Section 5.1). For example, the sentence (32) can be treated as either a sentence-based sequence (33) or two clause-based sequences (34). The segmentation of clauses prevents linguistically odd sequences, such as `OBJA-SUBJ-OBJA` (cf. (33-b)), from being learned in tagging models. To automatically segment clauses, I use topological field tags returned by the supertagger (cf. Chapter 4).

(32) Der Kantonsrat kann in den Bereichen Organisation und Personal Verordnungen erlassen, soweit das Gesetz dies vorsieht.

(33) Sentence as sequence

a. Input tokens:
Kantonsrat, Bereichen, Organisation, Personal, Verordnungen, Gesetz, dies

b. Output labels:
`SUBJ PN APP CJ OBJA SUBJ OBJA`

[12] Double object can occur in German - in this case, we have OBJA2

(34) Clause as sequence

a. (i) Input tokens:
Kantonsrat, Bereichen, Organisation, Personal, Verordnungen

(ii) Output labels:
`SUBJ PN APP CJ OBJA`

b. (i) Input tokens:
Gesetz, dies

(ii) Output labels:
`SUBJ OBJA`

5.3.1.2 Features

Features are key linguistic indicators that may be useful in the prediction of dependency grammar labels. Both the structured perceptron and the CRF algorithms are based on an indicator feature function. The indicator feature function assigns the existence (value = 1) or absence (value = 0) of the combination of linguistic cues in the current token and its contexts and the targeted tag of the current token (cf. Section 5.2).

For example, the first feature function (5.1), $f_1(cue, tag)$ states that the scalar of the feature vector is 1 if the condition 'the suffix of the current word is *-s* and target tag is `GMOD` (genitive object)' is true. It is assigned 0 if the condition is not fulfilled. The second feature function (5.2) states that the feature value is 1 if the previous token is a preposition and the current token is the dependent of a preposition (`PN`). It is assigned 0 if the condition is not true. In the example (35), the tokens *Bundesrechts* and *Völkerrechts* score 1 for the first feature and 0 for the second feature, whereas the token *Bestimmungen* scores 0 for the first feature and 1 for the second feature.

$$f_1(cue, tag) = \begin{cases} 1 & \text{if current word's suffix is } \textit{-s} \text{ and target tag is } \texttt{GMOD} \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

$$f_2(cue, tag) = \begin{cases} 1 & \text{if the previous token is preposition and target tag is } \texttt{PN} \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

(35) Die Kommission vollzieht die Entschädigungsabkommen nach den *Bestimmungen* dieser Abkommen sowie den anderen Vorschriften des *Bundesrechts* und den allgemeinen Grundsätzen des *Völkerrechts*.

Based on the analysis of word order in German presented in Section 5.1, I define 12 baseline features of sequential models. Table 5.2 provides an overview of these features, which are morphological, syntactic, semantic, and pragmatic linguistic cues for the linearization of grammatical functions and head-modifier dependency grammar relations (27). To extract features automatically, I use morphological features and topological dependency tags returned by the rule-based supertagger (cf. Chapter 4) and lemmas returned by the TreeTagger.

#	Linguistic cue	Context window	Description
1	Word form	[-2] [-1] [0] [+1] [+2]	Word form
2	Word character and suffix character	[0]	Capitalization, numeric characters, alphanumeric characters and two, three, four suffix characters of full nouns
3	Child string	[0]	(1) Article e.g. *der, die, das, eine, mein...* or the ending of strongly declined adjectives (1 or 2 characters, *-e, -em, -er, -es, -en*) or (2) the word forms of complements of prepositions
4	Child type	[0]	(1) The types of dependents of full nouns: Definiteness (i.e. definite, indefinite) or adjective, numeral or bare (2) The types of dependents of prepositions: full nouns, pronouns or relative pronouns
5	Coarse POS	[-1] [0] [+1]	Full nouns, pronouns, relative pronouns or prepositions
6	Topological fields	[-1] [0] [+1]	Vorfeld, mittelfeld, nachfeld or without field
7	Animacy	[0]	Person, organization or animal
8	Brown clustering	[-1] [0] [+1]	Top 4 and full Brown word clusters (2 features)
9	Contexts	[-1] [+1]	Coordinating conjunctions, comparative conjunctions, punctuations (i.e. comma, semicolon, colon), quotation mark, left and right brackets (`LB, RB, LXCOMP, RXCOMP`) and adverbs such as *nicht*
10	Predicate	[0]	Predicates
11	Voice	[0]	Passive or active voice

TABLE 5.2: Baseline features for ML-based supertagging (context [0] = current token, context[-1] = previous token, context[+1] = next token)

The following subsections describe each feature in detail.

Word Word forms are integrated with two features (cf. (36)). Word forms in the following positions are integrated as the first feature (36-a): current tokens ([0]), the previous token ([-1]), the token beforehand ([-2]), the next token of the current token ([+1]), and the next token afterwards ([+2]). In addition, three types of strings of current words, that is, capitalization, numeric, or alphanumeric strings, are captured in the second feature (36-b). Hence, the alphanumeric character of *1a* in a sequence *Artikel 1a*, for example, is provided as the cue for the prediction of an apposition `APP`.

(36) a. Word form (cf. #1 in Table 5.2)
b. Word character (cf. #2 in Table 5.2)

Morphology To capture the morphosyntactic properties of the nouns (i.e. case, number and gender) exposed in morphological marking, I created two features (37). To capture the suffix of a full noun, the last two, three, and four characters are extracted as the first feature (37-a). For example, the word *Allgemeinheit* has four features *-heit*, *-eit*, *-it* and *-t* and the word *Bundes* *-ndes*, *-des*, *-es* and *-s*. The suffix features capture some of the morphosyntactic properties of a full noun. In the example *Allgemeinheit*, *-heit* is a cue for singular and feminine morphosyntactic properties, whereas *-es* and *-s* in the example *Bundes* capture the morphosyntactic features of singular and genitive. Because the suffix of a noun is highly ambiguous, the morphological cues of the dependents of full nouns are extracted as second morphological features (37-b). The determiners or the inflectional endings of strongly declined adjectives exhibit morphosyntactic properties more distinctively than nouns do. Determiners such as *der* or *die* or the endings of strongly declined adjectives, such as *-em* or *-es*, are extracted as features based on POS tags. For example, a morphological feature *-em* of *gesamtschweizerisch**em*** in the sequence *gesamtschweizerischem Interesse* is extracted as the feature. It is unambiguously dative and a strong indicator that the head noun *Interesse* is an indirect object. However, there are some issues of morphological syncretism, such as *-e*. The inflection *-e* (e.g. *gerecht**e** Behandlung*) can be either nominative or accusative, that is, an indicator of a subject or a direct object. To extract the morphological features, I used the morphological features and dependency grammar labels returned by the rule-based supertagger.

(37) a. Morphological marking of nouns (cf. suffix character #2 in Table 5.2)
b. Morphological marking of dependents of nouns (cf. child string #3 (1) in Table 5.2)

Syntax As syntactic indicators, I created two types of features as (38). In the coarse POS feature (38-a), nouns are categorized into coarse part-of-speech tags, that is, full nouns, pronouns, relative pronouns, and truncations. As discussed in Section 5.1, the order of arguments varies depending on the pronominality. For example (39), the pronouns *sie* and *dies* are both ambiguous between the nominative and the accusative. However, by providing the information that the previous token is a pronoun and the current token is a pronoun, the second pronoun *dies* is less likely to be deemed nominative, following the canonical word order of pronouns (cf. Section 5.1). The topological fields feature (38-b) indicates the linear position of nouns and prepositions in a clause, that is, vorfeld, mittelfeld, and nachfeld. For example, in German, the vorfeld is a topic position that is occupied mainly by subjects. In the mittelfeld, there can be canonical word order but it can be also relatively freely scrambled (cf. Section 5.1). To extract the morphological features automatically, I used the morphological features and topological dependency grammar labels provided by the rule-based supertagger.

(38) a. Coarse POS (#5 in Table 5.2)
b. Topological fields (#6 in Table 5.2)

(39) Hat die ersuchende Stelle Grund anzunehmen, dass bereits ein Sicherheitsrisiko besteht, oder hat sie Kenntnis von einem Strafverfahren gegen die zu prüfende Person, so meldet **sie dies** schriftlich der zuständigen Prüfbehörde.

Animacy Animacy is the semantic classification of entities with regard to agentness. The traditional classes of animacy are person, animal, and thing. Because our texts are law texts, organizations are included in the animacy class. Organizations such as authorities or institutions often are personalized and are the agent of an action. As discussed in Section 5.1, animacy is a relevant indicator for the encoding of syntactic functions of arguments. The class *person*, for example, indicates a strong grade of agentness and is more likely to be a subject than a direct object (cf. Section 5.1). Moreover, animate entities tend to precede inanimate ones (cf. Section 5.1).

Two animacy features are captured as (40) depending on the extraction method used. The first feature (40-a) is based on the resources of animacy classes, *person*, *organization*, and *animal*. The second feature (40-b) is based on clustering classes.

(40) a. Resource-based animacy (Animacy #7 in Table 5.2)
b. Clustering-based animacy (Brown clustering #8 in Table 5.2)

For the first feature (40-a), three animacy classes are identified by using GermaNet (Kunze and Lemnitzer, 2007). I used the section *Mensch* ('people'), section *Gruppe* ('group') and section *Tier* ('animal'), respectively. To complement personal names, the output from the German morphological analyzer Gertwol (Haapalainen and Majorin, 1994) was added. As a supplement, I used a list of professions (e.g. *Bäcker/Bäckerin*) for the acquisition of entities denoting persons. The list contains 37,494 entities.[13]

For the second feature of animacy (40-b), the hierarchical classes of Brown clustering (Brown et al., 1992) are exploited. Brown clustering is a bigram-based word clustering that has been successfully integrated to improve parsing (Koo et al., 2008), domain adaptation (Candito et al., 2011), and named entity recognition (Miller et al., 2004). I ran the Brown clustering on the lemmatized tokens of TüBa data and Swiss German-language law texts to build 100 clusters.[14] The resulting clusters contained a great deal of animacy and named entities, as shown in Table 5.3. For the animacy features, I used the first four digits and all digits in the clustering hierarchy. They captured animacy the most clearly in the data.

[13] I obtained the list created by the Swiss Federal Statistical Office.

[14] I used the Brown clustering implemented by P. Liang: `https://github.com/percyliang/brown-cluster` (last visited on 2013/10/30)

Bit class	Entities	Types
0010	Departmente, Kanton, Gemeinde, Polizei, Justiz, ETH, Beamt, Professorin, Ratsmitglied, Arzt, Alpaca, Esel, Lama	Authorities, professions, animals
00110	Kuh, Ziege, Schwein, Katze, Hauskatze, Frosch, Fisch, Biene, Mücke	Animals
0011110	Verfassung, Verordnung, Deklarationsverordnung, Chemikalienverordnung, Zollverordnung, Störfallverordnung, Eisenbahnverordnung	Law
0100	Parlamentarier, Zollwert, EU-Kommissar, Nato-Generalsekretär, Grüner, Tänzer, Hausarzt, Löwe, Schäferhund, Hund	Professions and animals
011010	Bundesamt, Departement, Zivilstandsamt, Ministerium, Versicherungsgericht, Bundesverfassungsgericht, Bundesgericht	Authorities
011100	Bundesgesetz, Militärgesetz, Edelmetallkontrollverordnung, Kombiverkehrsverordnung, Rechtshilfegesetz	Law
011111	Fürsorgebehörde, Datenschutzbehörde, Flugunfalluntersuchungsbehörde, Zuteilungsorganisation, Jagdbehörde, Zivilschutzbehörde	Authorities
11110100	Volkswagen, Daimler-Benz, Suhrkamp, Siemens, ABB, Telecom, Schröder, Clinton, Thomas, Klaus, John, Paris, München, Frankfurt, London, Moskau	Company, person names and location
11110101	Henry, Tony, Herald, Roberto, Erhard, Theo, Jacques, Gerhard, Gregor, Marcus, Stephen, German, Miguel, Margaret, Laurent, Ingrid, Annemarie	Person names
111101101	Erika, Gabriele, Diane, Heidi, Evelyn, Tamara, Rosemarie, Philip, Jacob, Ullrich, Claus-Dieter, Gustaf, Muhammed	Person names

TABLE 5.3: An example of the Brown clustering hierarchy

Definiteness As a feature, the types of dependents of nouns are categorized into definiteness, indefiniteness, or none (i.e. adjective, numeric, bare nouns) (cf. child type #4 (1) in Table 5.2). The definiteness of a noun is relevant for the linearization of syntactic functions of arguments, as discussed in Section 5.1. Definite nouns tend to precede indefinite nouns. Furthermore, bare nouns, for instance, exhibit a cue for direct objects in light verb constructions (e.g. *Anwendung [finden]*) and for appositions (e.g. *[Kanton] Zürich*). To extract the feature automatically, I used the morphological features provided by the rule-based supertagger.

Contexts Because only nouns, pronouns, and prepositions are input tokens, those that are not input tokens are integrated as context features (cf. #9 in Table 5.2). The

context features are categorized into four types:

(41) a. Punctuations (comma, semicolon, colon or quotation mark)
b. Conjunctions (coordinating conjunction or comparative conjunction)
c. Brackets of clauses (right and left brackets of a clause)
d. Adverbial words such as *nicht, sogar, nur*

Punctuation (41-a) and coordinating conjunctions (41-b) are relevant cues for coordination (`KON` or `CJ`). Comparative conjunctions (41-b) are cues for the dependents of comparative conjunctions (`KOM`). The clause bracket feature (41-c) is used to specify the tokens adjacent to verbs. In particular, the tokens immediately followed by a clause bracket are often direct objects in light verb constructions (e.g. *Rechnung tragen*). In addition, adjacent adverbs are lexically integrated into the feature (41-d) to capture focus particles such as *nur* or *nicht*. To extract the feature automatically, I used the morphological features and topological dependency tags provided by the rule-based supertagger.

Predicate and Voice Predicates are included as features (#10 in Table 5.2) in order to capture the argument structure of a clause. The lemmata of predicative adjectives and full verbs are lexically integrated into features as (42). Passive or passive-like constructions are marked differently to indicate the alternation of the realization of the argument structure as (43). In addition, the voice of predicates is added as a feature (#11 in Table 5.2) to model passive vs. active constructions directly. To extract the feature automatically, I used the lemmas returned by TreeTagger, and morphological features and topological dependency tags provided by the rule-based supertagger.

(42) a. Present: e.g. *planst* → planen
b. Past: e.g. *geplant haben* → planen
c. Future: e.g. *planen werden* → planen
d. Modal: e.g. *planen müssen* → planen
e. Copula: e.g. *A B sein* → sein
f. Predicative adjectives: e.g. *notwendig sein* → notwendig

(43) a. Passive constructions: e.g. *geplant werden* → werden%passiv%planen
b. Zustand passive: e.g. *geplant sein* → sein%planen
c. Zu-infinitive *sein* passive: e.g. *zu planen sein* → sein%zu%planen

Prepositions Prepositions are distinguished between complements (`OBJP`) and adjuncts (`PP`) in the dependency tag set (cf. Table 5.1). In this distinction, the features (44) are considered useful. In the example sentence *Wir essen Pizza mit dem Gabel*, the predicate *essen*, the preposition *mit*, the dependent of the preposition *Gabel*, and the immediately preceding noun *Pizza* are lexically and syntactically integrated into the model. These features were used in the automatic recognition of PP attachments in a supervised approach in Kübler et al. (2008). Buchholz (1998) used contexts and syntactic features similar to (44) in memory-based learning in the automatic acquisition of subcategorization frames for lexicons.

(44)
a. Current word: e.g. *mit* (#1 Word form [0] in Table 5.2)
b. Previous word: e.g. *Pizza* (#1 Word form [-1] in Table 5.2)
c. The POS of the previous word: e.g. full noun (*Pizza*) (#5 Coarse POS [-1] in Table 5.2)
d. The child word of the current word: e.g. *Gabel* (Child string #3 (2) in Table 5.2)
e. The child POS of the current word: e.g. full noun (*Gabel*) (Child type #4 (2) in Table 5.2)
f. Predicate: e.g. *essen* (#10 in Table 5.2)

In summary, this section defined the linguistic unit of a sequence and a set of linguistically motivated features to predict the dependency grammar labels of nouns and prepositions. Based on this baseline, in the next section, I test the training of sequential models.

5.3.2 Experiments

In this section, I describe my experiments to train sequential models for the domain of Swiss German-language law texts. The objective of the following experiments is to develop a method for predicting dependency grammar functions with high accuracy, so they can be used in the style error detection presented in Chapter 2. This method should take into account the constraint that only minimal in-domain manual data annotation can be undertaken.

The remainder of this section is organized as follows: After I describe the data, I build the experiments incrementally as follows: the training and testing of (i) sequential models (i.e. the structured perceptron and CRF), (ii) the linguistic units of a sequence, (iii) combinations of features, and (iv) combinations of in-domain and out-of-domain training data.

5.3.2.1 Data

A large amount of training data is required to train the supervised ML models. Because no annotated in-domain law texts are available, I manually annotated 500 randomly selected sentences from the corpus of Swiss German-language law texts with dependency labels for nouns and prepositions by following the guidelines for dependency grammar (Foth, 2005) and the annotation of TüBa dependency conversion. These 500 sentences are not the same data set as the one used for the evaluation of the rule-based supertagger in the previous chapter. The 500 sentences were divided into 300 sentences (5,037 tokens, called 'Law300Train') for training and 200 sentences (5,042 tokens, called 'Law200Test') for testing. As 300 sentences are too few for training, I use TüBa-D/Z (henceforth: TüBa). TüBa is a linguistically annotated corpora of the newspaper, Die "Tageszeitung" (Taz). In the experiments, I divided the TüBa data into a training set (60% of TüBa, 700,888 tokens, named 'TüBa60Train'), a cross-validation set (20% of TüBa, 231,277 tokens), and a test set (20% of TüBa, 232,561 tokens, named 'TüBa20Test'). In the experiment, TüBa data were treated as out-of-domain data, while Swiss German-language law texts represented in-domain data.

Before the training and testing of the sequential models, nouns, pronouns and prepositions were augmented with automatically extracted features and then extracted into sequences. In this preprocessing, the existing annotation was used for TüBa data, and the automatic annotation processed by the rule-based supertagger was used for the law texts.

5.3.2.2 Model Selection

To compare tagging methods, I trained the structured perceptron algorithm and the CRF model with the baseline features presented in the previous section (cf. Table 5.2) on the TüBa training set (TüBa60Training). I re-implemented the algorithm of the structured perceptron described in Collins (2002). As reported (e.g. Nguyen and Guo, 2007), the structured perceptron reaches optimal performance after less than 10 epochs of learning. I trained the models of the structured perceptron in 10 learning iterations. In the CRF, I used the implementation of wapiti.[15] Cross-validation data were used to determine the best model in the perceptron and to reach the optimal convergence in CRF.

[15]Wapiti is available at: `https://wapiti.limsi.fr` (last visited on 2015/06/01)

I tested the trained models on the TüBa test data (TüBa20Test). The best perceptron model (ninth learning epoch) achieved a label accuracy of 80.83% and CRF achieved a label accuracy of 89.59%. Label accuracy is calculated by dividing the number of correctly predicted tags by the total number of the tokens (cf. Subsection 4.2.4.2). The best model of the perceptron was the averaged model, which confirmed the results of Collins (2002). Roark et al. (2004) compared the structured perceptron and CRF in speech recognition. CRF performed better in parameter estimation than the structured perceptron did. Similar results were reported by Wang et al. (2006). Nguyen and Guo (2007) tested several sequential models for POS tagging and Optical Character Recognition (OCR). The results showed that structured SVM (Tsochantaridis et al., 2005) had the best results in both tasks. In POS tagging, the perceptron was competitive with the best model. In contrast, CRF was one of the worst models in both tasks. This finding contradicts the results reported by Wang et al. (2006), Roark et al. (2004). Thus, the performance of machine learning methods differs depending on the task. In my task, CRF outperformed the structured perceptron.

In conclusion, I found that CRF outperformed the structured perceptron in the task. I use only the CRF method in the remaining experiments.

5.3.2.3 Sequence Selection

In NLP, a sequence is typically considered a sentence. In tagging the syntactic categories of arguments, however, linguistically adequate sequences are clauses, as discussed in Section 5.3.1.1. Computationally, clauses are much shorter than sentences, which speeds up the normalization in CRF in training and testing. To determine the optimal sequence for CRF, I trained CRF models on the TüBa training set (TüBa60Training) based on sentence-based and clause-based sequences with the baseline features. Clauses are chunked based on topological fields assigned by the rule-based supertagger (cf. Chapter 4).

I tested the two models on the TüBa test set (TüBa20Test) and the Swiss law test set (Law200Test). Table 5.4 illustrates the label accuracy of dependency grammar labels and the average label accuracy with regard to the arguments in brackets. Regardless of text type, the clause-based CRF model outperformed the sentence-based CRF model. On the test set of TüBa, the clause-based model was significantly better than the sentence-based model.[16] In the recognition of syntactic functions of arguments, the clause-based

[16]The Fisher's test was applied to the law texts. As the TüBa data is large enough, chi-square with Yates correction was used for the significance test.

models also outperformed the sentence-based models.

#	Feature	Sequence	Training	Test: Law200Test	Test: TüBa20Test
1	Baseline	Clause	TüBa60Train	**84.56**% (82.12%)	**89.59**% (79.94%)
2	Baseline	Sentence	TüBa60Train	82.77% (76.31%)	88.82% (78.57%)
				p=0.1303	$p < 0.0001$

TABLE 5.4: Label accuracy: sentence-based vs. clause-based supertagger

In conclusion, the results showed that the clause-based CRF model outperformed the sentence-based CRF model. In the remaining experiments, the sequences are defined as clauses.

5.3.2.4 Feature Combination

I combined the basic features (cf. Table 5.2) to create new features. The combined features are tightly integrated into the conditions for probability estimation. For example, the topological fields feature (#6 in Table 5.2) and the animacy feature (#8 in Table 5.2) were combined to build a feature (5.3). Or, the next feature (5.4) is created by combining the coarse POS feature (#5 in Table 5.2) of the current token and the topological field feature (#6 in Table 5.2) of the previous token.

$$f_{100}(cue, tag) = \begin{cases} 1 & \text{if the current word is placed at vorfeld and denotes person,} \\ & \text{and the target tag is subject} \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

$$f_{101}(cue, tag) = \begin{cases} 1 & \text{if the current word's POS is full noun and the previous token is} \\ & \text{placed in vorfeld, and the target tag is direct object} \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

In this experiment, the baseline features were combined in three ways (cf. (45)). In the first feature combination (45-a), all two basic features were combined. In the second feature combination (45-b), the two features were combined if they belonged to the same

context window. The 100th feature (cf. (5.3)) is an example of this type of feature combination. In this feature, the two features of the current token are combined. The 101th feature (cf. (5.4)) does not belong to the feature set (45-b). In the third feature combination (45-c), the two features were combined if they belonged to the same linguistic representation. The 101th feature (cf. (5.4)) is an example of this feature combination. The POS and topological fields belong to the same linguistic representation, i.e. syntax (cf. (45-c-ii)). In contrast, the 100th feature (cf. (5.3)) does not belong to the feature set (45-c).

(45)
- a. All two features are combined (cf. #3 in Table 5.5)
- b. Two features in the same context windows are combined (cf. #4 ibid.)
- c. Two features that belong to the same linguistic representations (i.e. morphology, syntax and semantic) are combined (cf. #5 ibid.)
  - (i) Morphology: 'word character and morphology'(#2 in Table 5.2), 'child string' (#3 ibid.) and 'child type'(#4 ibid.)
  - (ii) Syntax: 'POS' (#5 ibid.), 'topological fields' (#6 ibid.) and 'voice' (#11 ibid.)
  - (iii) Semantic: 'animacy' (#8 and #9 ibid.)

I trained the CRF model with these three feature sets on the TüBa training set (TüBa60-Train). I then tested the trained models on the test sets. The combination of the same context window features showed the best performance on TüBa. The improvement was significant. However, none of the feature combinations showed significant improvement for law texts, as shown in Table 5.5.

#	Feature configuration	Training	Test: Law200Test	Test: TüBa20Test
1	Baseline	TüBa60Train	**84.56**% (82.12%)	**89.59**% (79.94%)
3	Baseline + all two-way combined baseline features	TüBa60Train	84.31% (82.72%)	**91.04%** (80.45%)
4	Baseline + two-way combined baseline features in the same context	TüBa60Train	**84.80**% (83.24%)	90.86% (**81.85%**)
5	Baseline + two-way combined baseline features in the same linguistic representation	TüBa60Train	84.65% (81.39%)	89.70% (79.33%)
			p=0.8631 (#1 and #4)	p <0.0001 (#1 and #3)

TABLE 5.5: Label accuracy: feature combinations for the CRF-based supertagger

In this experiment, I trained and tested three combinations of the baseline feature set. One of these feature combinations improved the performance on TüBa, but there was no significant improvement on the law texts. Because the objective of the experiment was to optimize the supertagger for law texts, I use the baseline features for the remaining parts of the experiments. In addition, the simple baseline feature set required much less computation time for training and testing.

5.3.2.5 Data Selection: Out-of-Domain Data

In this subsection, I explore how the existing large amount of annotated out-of-domain training data (here: TüBa60Train) could be adapted to the legislative domain. I use two different types of data selection methods:

(46) a. Cross-entropy based data selection
b. Linguistically motivated data selection

Cross-Entropy-Based Data Selection In cross-entropy-based data selection, sentences that are similar to the target legislative domain are automatically extracted from a large amount of out-of-domain data. Cross entropy is a variant of perplexity that is used to compare different probability models. It is measured as follows (Jurafsky and Martin, 2009, pp. 117):

$$H(w_1 \ ... \ w_n) = -\frac{1}{N} logP(w_1 \ ... \ w_n) \tag{5.5}$$

Cross entropy is approximated by computing the log probability of a sequence $w_1 \ ... \ w_n$, normalized by the length of the sequence. In this experiment, word-based and POS-based language models were trained on in-domain and out-of-domain language data.[17] I used the language model, Berkeley LM (Pauls and Klein, 2011).[18] I trained 4-gram language models with Kneser-Ney back-off on the TüBa training data (39,313 sentences, 700,888 tokens) and the same amount of law text data (25,901 sentences, 700,974 tokens).

To measure the similarity of out-of-domain data to the targeted in-domain legislative data, I used the following two measurements:

(47) a. Ranking-based entropy score
b. Difference-based entropy score

Ranking-based entropy scoring (47-a) is a measurement of how surprisingly out-of-domain sentences are encountered based on the experiences collected in the target-domain data, that is, the language model trained on target-domain data. The cross entropy scores were ranked by ordering from low to high. Out-of-domain TüBa sentences were assumed similar to the targeted legislative domain data if they were low in cross entropy, that is, less surprising. This method is compatible with Axelrod et al. (2011) in which perplexity was used instead of cross entropy.

Difference-based entropy scores (47-b) are a measurement of differences in entropy scores for out-of-domain sentences returned by a language model trained on target-domain data and by a language model trained on out-of-domain data. Out-of-domain TüBa sentences were considered similar to target-domain sentences if the difference of entropy scores between those two models was small. The method is based on Moore and Lewis (2010).

In the experiment, I trained CRF models on 25, 50, 75, 85, and 95% of the TüBa training set. I tested the models on the test set of the law texts. The results are shown in Table 5.6. A striking result was that 25% of the training data were competitive with 100% of the training data with regard to overall label accuracy. With regard to grammatical functions, the model trained on 25% of the training data selected by

[17] POS tags were already annotated in TüBa. For the law texts, they were converted from the morphological analyses provided by the rule-based supertagger.
[18] Berkeley LM is available at: `https://code.google.com/p/berkeleylm/` (last visited: 2015/03/09)

POS-based difference-entropy scores was competitive with the model trained on 100% of the training data. In this experiment, the TüBa training data (25%) selected by the POS-based difference model (#21 in Table 5.6) was the best training data for law texts because the trained CRF model was balanced in label accuracy and the recognition of syntactic functions.

#	Training		Test:	
#	Selection	% of TüBa60Train	Law200Test	
1	None	100%	**84.56**%	(82.12%)
6	Word-Entropy-Ranking	25%	84.17%	(75.47%)
7		50%	83.74%	(77.05%)
8		75%	84.51%	(75.22%)
9		85%	84.22%	(83.02*%)
10		95%	83.83%	(82.25*%)
11	Word-Entropy-Difference	25%	84.85*%	(63.20%)
12		50%	84.17%	(62.15%)
13		75%	84.36%	(83.18*%)
14		85%	**84.94***%	(83.32*%)
15		95%	84.80*%	(83.30*%)
16	POS-Entropy-Ranking	25%	83.83%	(82.43*%)
17		50%	84.27%	(82.35*%)
18		75%	83.93%	(81.43%)
19		85%	84.31%	(81.66%)
20		95%	84.80*%	(83.20*%)
21	POS-Entropy-Difference	25%	84.89*%	(**83.40***%)
22		50%	84.51%	(83.17*%)
23		75%	84.41%	(82.61*%)
24		85%	84.56%	(82.71*%)
25		95%	84.36%	(82.16*%)

TABLE 5.6: Label accuracy: automatic in-domain data selection for the CRF-based supertagger (baseline feature)

To gain insights into the best training data in this experiment section (#21 in Table 5.6), I computed the distribution of sentence lengths and compared it with the one of the entire TüBa training set (TüBa60Train). Table 5.7 illustrates that the entropy-based training data contains a higher proportion of long sentences than the full TüBa training set. Therefore, the distribution of sentence lengths in the entropy-based selected training data is more similar to Swiss German-language law texts presented in the previous chapter (cf. Table 4.30 and the summarization in Table 5.7). In Swiss German-language law texts, long sentences are much more frequent than in the full TüBa training data.

Sentence length:	**TüBa60Train**		**POS-Entropy-Diff. 25%**		**Swiss G-lang. law**	
# of tokens	**# of sent.**	**(%)**	**# of sent.**	**(%)**	**# of sent.**	**(%)**
1–14	17,317	44	3,942	40	47,543	29
15–24	12,621	32	3,252	33	55,425	34
25–34	6,154	16	1,699	17	30,629	19
35–44	2,124	5	599	6	13,988	8
45–54	706	2	227	2	6,699	4
≥ 55	391	1	106	2	10,480	6

TABLE 5.7: TüBa training data sets (#1 and #21 in Table 5.6) and Swiss German-language law texts (Swiss G-lang. law): Proportion in sentence length

Linguistically Motivated Data Selection Compared with the entropy-based automatic selection of the training data previously presented, I conducted an experiment with a TüBa training data set that was selected based on the following linguistic observations: the law texts did not contain verbless clauses, fragmental clauses, or parenthetical clauses. In such partial clauses, the dependency relations `ROOT` and `PAR` were assigned to nouns or prepositions. In the experiment, I removed clauses from the TüBa training data (`TüBa60Train`) that contained dependency labels `ROOT` or `PAR` were assigned to nouns or prepositions. The selected training set contains 56,296 clauses, which is 89% of the whole TüBa training set (TüBa60Train).

I trained a CRF model with the baseline features on this newly created TüBa data, which is called TüBa60NoLawNoise (cf. #26 in Table 5.8). Table 5.8 shows the results. The model trained on these linguistically selected training data outperformed the model trained on 100% of the TüBa training data set (#1 in Table 5.8) and the best one trained on the data based on the entropy-based training data selection (#21 in Table 5.8).

#	Feature	Training	Test: Law200Test
1	Baseline	TüBa60Train	84.56% (82.12%)
21	Baseline	POS-Entropy-Difference 25% of TüBa60Train	84.89% (83.40%)
26	Baseline	TüBa60NoLawNoise	**85.81**% (**83.73**%)
			p= 0.2743 (#1 and #26)

TABLE 5.8: Label accuracy: linguistically motivated in-domain data selection for the CRF-based supertagger

In conclusion, this subsection described two methods of selecting the out-of-domain training data for the target domain. Both the entropy-based and the linguistically motivated training data selection were slightly better in the accuracy of the trained

models than the model trained on 100% of the out-of-domain data was. Because the former method resulted in better accuracy than the latter method did, and because smaller models are preferable, I use the linguistically motivated selected TüBa training set (TüBa60NoLawNoise) for the remaining experiments.

5.3.2.6 Data Selection: In-Domain Data

In this subsection, I explore to which degree manually annotated in-domain data improves the performance of the CRF-based tagger.

I trained CRF models on the TüBa training set for the law texts (TüBa60NoLawNoise) combined with three in-domain data sets - 100 sentences ('Law100Train' in #27 of Table 5.10), 200 sentences ('Law200Train' in #28 of Table 5.10) and 300 sentences ('Law300Train' in #29 of Table 5.10).

Table 5.9 shows that the performance was improved by adding a small amount of in-domain data. Only 300 sentences of in-domain annotation data improved the label accuracy of approximately 1% of the data in the previous experimental setting, that is, only the out-of-domain training data.

#	Feature	Training	Test: Law200Test
26	Baseline	TüBa60NoLawNoise	**85.81**% (83.73%)
27	Baseline	TüBa60NoLawNoise + Law100Train	86.00% (82.73%)
28	Baseline	TüBa60NoLawNoise + Law200Train	86.15% (83.03%)
29	Baseline	TüBa60NoLawNoise + Law300Train	**86.53**% (83.47%)

TABLE 5.9: Label accuracy: Addition of in-domain data for the CRF-based supertagger

In this section, I experimented on two statistical tagging methods (the structured perceptron and the CRF), input data (sequences and features), and the training data (in-domain data and out-of-domain data). The task was to assign dependency grammar labels automatically to nouns and prepositions. In this task, the CRF-based tagger performed better than the structured-perceptron based tagger. The best CRF model was trained on (i) the data with clause-based sequences and the baseline features and (ii) the out-of-domain TüBa training data with the linguistically motivated data selection, combined with a small number of manually annotated in-domain sentences. In the next section, the best CRF model is combined with the rule-based supertagger (cf. Chapter 4) and state-of-the-art parsers for further optimization.

5.4 Hybrid Supertagging

Despite the careful configuration of CRF models, the CRF-based supertagger is still limited because the prediction is based on the probabilistic account of information on current tokens and the contexts. Thus, we need to integrate the principle-based hard linguistic constraints into the probabilistic model. In this section, the CRF-based supertagger is combined with the rule-based supertagger (cf. Chapter 4). The rule-based supertagger provides the hard linguistic constraints. Additionally, I consider combining it with state-of-the-art parsers in order to integrate their different views.

Combining systems is not new. In tagging and parsing, two types of combination have been proposed: 1) division of labor and 2) ensemble:

(48) a. Division of labor: *More than one system are combined in a sequential manner. Each system has a different task.*
b. Ensemble: *More than one system is combined in a parallel manner. Each system has the same task.*

Division of labor (48-a) is typically introduced as a pipeline architecture of parsers based on the divide-and-conquer philosophy. For example, in a German parser developed by Hinrichs et al. (2002), a finite-state, rule-based topological field chunking component and history-based parsing were sequentially processed. In Foth and Menzel (2006), a rule-based weighted constraint dependency parser was combined with statistical components, that is, TreeTagger chunking, a trigram TnT supertagging, a probabilistic model of PP-attachment, and a shift-reduce statistical parser.

In the second type of combination (48-b), more than one parser or tagger work in parallel on the same task. The outputs returned by them are then aggregated and combined by means of (A) feature, (B) meta classification, or (C) voting. The first type (A) is an ensemble system in which one system guides another system by being integrated as a feature. For example, in a parser developed by Nivre and McDonald (2008), a graph-based and a transition-based dependency parser were combined by allowing one model to be guided by the other one as a feature. Both guided parsing models improved the standalone parsers. The second type of ensemble system (B) builds a classifier to select the best output among the multiple outputs generated by various systems. For example, Henderson and Brill (1999) combined three constituent parsers by building a meta-classifier based on Naive Bayes. This approach outperformed the accuracy of standalone parsers. In the third type (C), we select the output if the majority of systems agreed. For example, Green and Žabokrtský (2012) combined graph-based and transition-based

dependency parsers with constituent parsers by voting. The results showed that the use of various parsers achieved the best score. Similar results were reported by Surdeanu and Manning (2010). In most cases, when CG-based taggers are combined, a statistical tagger is sequentially introduced for the final selection of the tags assigned previously by the CG-based taggers (e.g. Hajič et al., 2001, Ezeiza et al., 1998, Tapanainen and Voutilainen, 1994).

For the combination of a CG-based supertagger and a CRF-based supertagger, we are interested using in systems that perform the same task (i.e. ensemble). Because the rule-based supertagger returns not only dependency grammar labels but also morphosyntactic case features, we focus on the combination of heterogeneous output representations. Because the voting method is generally applicable to homogeneous representations, the choice is between a feature-based method or a meta-classifier method. In a feature-based approach, the morphosyntactic case features and dependency grammar labels returned by the rule-based supertagger can be weighted and incorporated into a sequential CRF model as features, that is, parallel to syntactic, semantic, and pragmatic features. In a meta-classifier method, heterogeneous outputs can be combined. However, each is handled as an autonomous feature in a classifier. In our case, the morphosyntactic hard constraints and dependency grammar relations returned by the rule-based supertagger and the dependency grammar relations returned by the CRF-based supertagger are the interacting factors considered in the decision about final dependency grammar relations, which is not the best solution from a linguistic point of view. Therefore, in the next section, I follow a feature-based approach and present a feature-based hybrid supertagger.

5.4.1 Hybrid Supertagger: CRF-Tagger Meets CG-Based Tagger

In this subsection, the CG-based tagger (cf. Chapter 4) was combined with the best CRF model as determined by the experiments (#29 in Table 5.9) in a feature-based method. I added information about morphosyntactic case and the dependency labels returned by the rule-based tagger as features into the in-domain training data (Law300Training).

I then trained a CRF model on this data, which was previously combined with the TüBa training set selected for the law texts (TüBa60NoLawNoise). I tested this newly trained model on the test set of the law texts. Table 5.10 shows the results. The tagger was improved by the integration of the case features (cf. #30 of Table 5.10) and dependency relation labels (cf. #31 of Table 5.10). By combining these two features, the CRF model achieved the best performance of all experiments (cf. #32 of Table 5.10).

#	Feature	Training	Test: Law200Test
29	Baseline	TüBa60NoLawNoise + Law300Train	86.53% (83.47%)
30	Baseline + case features	TüBa60NoLawNoise + Law300Train	88.61% (87.96%)
31	Baseline + dependency label from the rule-based tagger	TüBa60NoLawNoise + Law300Train	92.57% (**90.75**%)
32	Baseline + case features + dependency label from the rule-based tagger	TüBa60NoLawNoise + Law300Train	**92.86**% (90.38%)
			p<0.0001 (#29 and #41)

TABLE 5.10: Test results (label accuracies) with various feature configuration and training data for the CRF-based supertagger

In summary, the addition of 300 in-domain sentences augmented with case features and output labels returned by the rule-based tagger significantly improved the tagger compared with the same training data without feature-based integration (92.86% vs. 86.53%).

Error Analysis To gain insight into the feature-based combination, I analyzed the test results for the CG-based tagger and the CRF-based tagger. I also analyzed the test results for the feature-based combination of these two taggers, that is, the hybrid tagger (cf. #32 of Table 5.10). Table 5.11 shows that the CG-based tagger, the CRF-based tagger, and the hybrid tagger achieved label accuracies of 84.20%, 86,53% and 92.86%, respectively. With regard to the main dependency labels illustrated in Table 5.11, the CRF-based tagger was improved by the guidance of the rule-based tagger. However, the hybrid tagger lowered the accuracy of the rule-based tagger with regard to coordination (`KON, CJ`).

	CG	CRF (#29)	Hybrid (#32)
Trained on	Without training	TüBa60NoLawNoise + Law300Train	TüBa60NoLawNoise + Law300Train
Features	-	Baseline	Baseline+case+label (CG-based tagger)
Tested on	Law200Test	Law200Test	Law200Test
Overall label accuracy	84.20% (1467, 275) [without OBJP and PP]	86,53% (1793,279)	92.86% (1924, 148)
SUBJ	0.82 (0.85, 0.81)	0.89 (0.91, 0.88)	**0.95** (0.96, 0.95)
OBJA	0.68 (0.67, 0.68)	0.72 (0.69, 0.74)	**0.87** (0.84, 0.90)
OBJD	0.34 (0.63, 0.24)	0.68 (0.73, 0.64)	**0.73** (0.75, 0.72)
GMOD	0.84 (0.80, 0.87)	0.88 (0.84, 0.93)	**0.95** (0.95, 0.95)
APP	0.69 (0.60, 0.82)	0.63 (0.53, 0.78)	**0.74** (0.66, 0.83)
KON	**0.96** (0.94, 0.99)	0.75 (0.78, 0.72)	0.90 (0.91, 0.90)
CJ	**0.95** (0.94, 0.95)	0.74 (0.81, 0.68)	0.91 (0.93, 0.91)
OBJP	- -	0.36 (0.45, 0.30)	**0.48** (0.63, 0.40)
PP	- -	0.93 (0.92, 0.95)	**0.94** (0.93, 0.95)

TABLE 5.11: Testing the models on law Texts: CG-based, CRF-based and hybrid tagger with regard to dependency labeling of nouns and prepositions

To measure the strength of the integration of the CG-based tagger (linguistic principle-based hard constraints) into the CRF-based tagger (linguistic probability-based soft constraints) and its effects on performance, I measured the degree to which the hybrid tagger was guided by the CG-based tagger. In the test data (Law200Test), 1512 tokens were nouns.[19] The hybrid followed the guidance of the CG-based tagger on 1,286 tokens (85% of nouns), 97.51% of which were correctly tagged, that is, the label accuracy was 97.51%. The hybrid tagger accepted suggestions from one of the labels provided by the CG-based tagger on 127 tokens (8% of nouns). In this case, the label accuracy was 96.85%. Two systems did not agree and predicted different tags on 99 tokens (7% of nouns). In this case, the label accuracy of the CG tagger was 40.74%, whereas that of the hybrid tagger was 35.18%. Thus, the label accuracy of the hybrid CRG-based tagger was high if it was guided by the CG-based tagger, and it selected the outputs suggested by the CG-based tagger. Correspondingly, the label accuracy of the hybrid tagger was low if the CRF model did not accept suggestions from the CG-based tagger. In this case, both the CG-based tagger and the hybrid tagger performed poorly (label accuracy of 40.74% and 35.18%, respectively). These results indicated that the poor accuracy of

[19] Of 1512 tokens, 560 tokens were prepositions. I excluded these tokens because the CG-based tagger does not provide dependency grammar relations for prepositions.

the CG-based tagger led to the CRF's rejection of the CG's suggestions. However, the alternative hybrid tagger was not very accurate (label accuracy of 35.18%).

Agreement of the hybrid tagger with the CG-based tagger	% of total number of nouns	Hybrid supertagger's accuracy
Exactly agreed	85%	97.51%
Partly agreed	8%	96.85%
Not agreed	7%	35.18%

TABLE 5.12: Agreement of the CG-based tagger and CRF-based tagger

In addition, I analyzed the confidence scores returned by the hybrid tagger (i.e. hybrid CRF model) and the accuracy of the hybrid model. In Table 5.13, we can observe that the average of the confidence scores is in concordance with the agreement and the correctness of the prediction. Thus, the confidence scores clearly indicated the performance of the hybrid tagger.

Agreement and Prediction	Average of confidence score
Exact match with CG and correct prediction	0.98
Exact match with CG and incorrect	0.89
match with one of CG outputs and correct	0.92
match with one of CG outputs and incorrect	0.10
Not agreed and only CRF correct	0.79
Not agreed and only CG correct	0.72
Not agreed and incorrect both	0.58

TABLE 5.13: Agreement, confidence score and prediction

Comparison To evaluate the performance of the hybrid supertagger, I compared it with two state-of-the-art parsers: the ParZu parser and the Bohnet parser. ParZu is a dependency parser based on a rule-based component combined with statistical components.[20] The Bohnet parser is a statistical parser in which a transition-based model or

[20] As preprocessing, a CRF-based POS tagger (clevertagger) and a morphological analyzer (Zmorge) are used. The parser is available at: https://github.com/rsennrich/parzu (last visited: 15/10/2015)

graph-based model can be trained.[21] We used the graph-based Bohnet parser in Chapter 2.

I trained a graph-based model and a transition-based model of the Bohnet parser on the training data of TüBa (TüBa60Train). I tested the models on the test set of law texts (Law200Test).[22] Table 5.14 provides an overview of the results. My hybrid supertagger outperformed all three parsers in label accuracy. The parser ParZu outperformed the state-of-the-art transition-based and graph-based parsers. In particular, it showed better accuracy than the hybrid supertagger (cf. #32 in Table 5.11) when it comes to direct objects (`OBJA`), indirect objects (`OBJD`), genitive modifiers (`GMOD`) and conjunctions (`CJ`).

	Transition-based Bohnet	Graph-based Bohnet	ParZu
Trained on	TüBa60Train	TüBa60Train	Off-the-shelf (Statistics: Europarl)
Tested on	Law200Test	Law200Test	Law200Test
Preprocessing	TreeTagger's POS and lemma	TreeTagger's POS and lemma	ZMorge and Clevertagger
Overall label accuracy	83.25% (1725, 347)	84.80% (1757, 315)	88.32% (1830, 242)
SUBJ	0.85 (0.93, 0.78)	0.85 (0.91, 0.79)	0.91 (0.94, 0.89)
OBJA	0.82 (0.78, 0.87)	0.79 (0.76, 0.82)	**0.89** (0.93, 0.85)
OBJD	0.56 (0.67, 0.48)	0.52 (0.65, 0.44)	**0.79** (0.75, 0.84)
GMOD	0.94 (0.96, 0.93)	0.93 (0.95, 0.91)	**0.96** (0.97, 0.95)
APP	0.55 (0.54, 0.57)	0.61 (0.79, 0.50)	0.66 (0.64, 0.68)
KON	0.43 (0.72, 0.31)	0.48 (0.74, 0.36)	0.44 (0.64, 0.33)
CJ	0.98 (0.97, 0.98)	0.97 (0.98, 0.96)	**0.99** (0.99, 0.98)
OBJP	0.19 (0.23, 0.17)	0.39 (0.48, 0.33)	0.52 (0.58, 0.47)
PP	0.92 (0.93, 0.91)	0.93 (0.94, 0.91)	0.94 (0.96, 0.91)
-	$p < 0.0001$ (CRF#32 and Bohnet)	$p < 0.0001$ (CRF#32 and Bohnet)	$p < 0.0001$ (CRF#32 and ParZu)

TABLE 5.14: Testing the models on law texts: ParZu, transition-based and graph-based Bohnet parser with regard to dependency labeling of nouns and prepositions

In this subsection, I presented the hybrid supertagger in which the morphosyntactic case features and dependency grammar labels returned by the rule-based supertagger were

[21] The various versions of Bohnet parser are available at: http://code.google.com/p/mate-tools/ (last visited: 15/10/2015). We used the version anna-3.6 for the graph-based parser and the version anna-3.3 for the transition-based parser. I trained these two parsers on the data with the first, second, and third order factors.

[22] The POS tagger, TreeTagger, is used in preprocessing. Because I annotated the law texts only with respect to the dependency labels of nouns and prepositions, I did not train the Bohnet parser on the training data of the law texts.

integrated as features into the CRF-based supertagger. This hybrid tagger improved the best CRF model, which was determined by the experiments presented in Section 5.3.2, in label accuracy. It also outperformed three state-of-the-art parsers. The results showed that the accuracy of the feature-based hybrid tagger was high if it followed the guidance of the rule-based tagger, and if the confidence score of the hybrid tagger was high.

5.4.2 Voting Supertagger: Hybrid Tagger Meets Parsers

Although the feature-based combination of the rule-based tagger and the CRF-based tagger achieved a high score of label accuracy, the results indicated that the combination did not exhaust these two different views.

In this section, I introduce a method in which various views were combined directly. Using a voting method, I combine these two supertaggers (i.e. CG-based and hybrid CRF-based tagger, henceforth: CRF-based tagger) with three state-of-the-art parsers: the ParZu parser and transition-based and graph-based Bohnet parsers. For the development of an ensemble supertagger, I used the test set used in the experiments (TüBa20Test) as the development set. I experimented on the following three types of voting systems:

(49) a. Simple voting
b. Confidence-based voting
c. Back-off voting

Simple Voting I build three simple voting systems based on Green and Žabokrtský (2012).

(50) a. Simple voting
b. Weighted voting
c. Ranked voting

In simple voting (50-a), each of five systems receives one vote. In weighted voting (50-b), each of five systems receives a voting score according to the label accuracies estimated in the development set (Law200Test) (cf. 5.11 and 5.14). Accordingly, CG, CRF, ParZu, the graph-based Bohnet parser, and the transition-based Bohnet parser had voting scores of 0.84, 0.93, 0.88, 0.85, and 0.83, respectively. In ranked voting (50-c), each system received a voting score following the ranking of the label accuracies evaluated in the

development set. The most accurate system (here the CRF-based supertagger) received five votes, whereas the least accurate system (here the transition-based Bohnet parser) received one vote. When the voting score was tied, the prediction of the best system (CRF) was chosen.

I tested these three voting systems on the development set. The results are shown in (51). All three voting systems underperformed the best standalone system, which was the CRF-based tagger (a label accuracy of 92.86%).

(51) a. Ensemble (50-a): 92.19% (1912, 162)
b. Ensemble (50-b): 85.10% (1765, 309)
c. Ensemble (50-c): 85.44% (1772, 302)

Confidence-based Voting As described in Söl ection 5.4.1, the confidence score of the CRF-based tagger was correlated to accuracy. In the three voting systems (52-a)-(52-c) defined below, the confidence score was used as the threshold for choosing the dependency grammar labels agreed by both the CG-based tagger and the CRF-based tagger. If the confidence scores were lower than the thresholds, or the CG-based tagger and the CRF-based tagger did not agree, the three simple voting methods (50) were introduced to gain consensus.

(52) a. CRF/CG agreement with a high CRF confidence score, otherwise simple voting
b. CRF/CG agreement with a high CRF confidence score, otherwise weighted voting
c. CRF/CG agreement with a high CRF confidence score, otherwise ranked voting

I set the threshold of an upper-bound confidence score at the third quartile of the confidence scores of the wrongly predicted tags. Hence, the majority of the instances that predicted wrongly went further to one of the three simple voting actions. At the same time, the majority of the correctly predicted instances were accepted as winners. If the CG-based tagger and the CRF-based tagger agreed exactly on one dependency grammar label, then the threshold was 0.99. If they partially agreed, then the threshold was 0.76. If the scores were tied, the prediction of the best system (CRF) was chosen.

I tested these three confidence-based voting systems on the development set. The results are shown in a list (53). The simple voting system with the use of confidence scores (label accuracy of 93.78%) slightly improved over the accuracy of the best standalone

CRF system (label accuracy of 92.86%). The other two systems underperformed the best standalone system.

(53) a. Ensemble (52-a): **93.78**% (1945, 129)
b. Ensemble (52-b): 92.38% (1916, 158)
c. Ensemble (52-c): 92.43% (1917, 157)

Back-off Voting In the error analysis in Section 5.4.1, I showed that the confidence scores generated by the CRF-based tagger, and the agreement of the rule-based tagger and the CRF-based tagger indicate the accuracy of the CRF-based tagger (cf. Table 5.13). The accuracy of the CRF-based tagger was low for edge cases, in which two systems did not agree (7% of test data set, cf. Table 5.12). Based on these results, I developed a back-off voting method: I combined the CG-based supertagger and the CRF-based supertagger as the priority. In the case when these two supertaggers agree with high confidence, the agreed tags are taken as winner. The confidence scores were used as the threshold in the same way as they were used in confidence-based voting (52-a)-(52-c). When they agree with low confidence, or they do not agree, the two parsers, the ParZu parser and the graph-based Bohnet parser, served as a back-off. ParZu was used first and then Bohnet was used second according to the label accuracies estimated on the development set. If voting tied the score, then the one with the highest F1 score won. In our case, ParZu's prediction won, if this parser returned the following dependency labels: `GMOD, OBJD, CJ, EXPL, OBJP`. Otherwise, the output of the CRF-based supertagger won. Compared with the confidence-based voting methods (52), this back-off voting method clearly prioritized the systems in the committee: The first priority was the CG-based and CRF-based supertagger as the principal committee. The next priority was the ParZu parser as the first back-off. The lowest priority was the Bohnet parser, which was introduced as the second back-off in the voting.

To build the back-off voting system, I considered the following four agreement cases:

(54) a. CRF-based and CG-based tagger agree
b. CRF-based and CG-based tagger partly agree
c. CRF-based and CG-based tagger do not agree
d. CG-based tagger does not predict

In the first case (54-a), the CRF-based and CG-based tagger agreed exactly on the tags. In the voting, they won if the confidence scores were higher than the third quartile of the wrongly predicted cases (0.99). Otherwise, they won if these two supertagger agreed with the first back-off system, the ParZu parser. If the first back-off did not agree with

them, then the second back-off system, the Bohnet parser, was compared to them. In the case of disagreement, the F1 scores of the committee members determined the winner. The back-off method was formulated as follows:

$$\begin{cases} CRF(x) = CG(x) & \text{if } CRF(x) = CG(x) \text{ and } Conf(x) \geq 0.99 \\ Vote(CRF(x), PARZU(x)) & \text{else} \\ Vote(CRF(x), BOHNET(x)) & \text{else} \\ F1(CG(x), CRF(x), PARZU(x), BOHNET(x)) & \text{else} \end{cases} \tag{5.6}$$

The function `system (x)` e.g. CRF(x) is the dependency tag assigned by the CRF-based tagger, given the input token x. The function `Conf(x)` maps the confidence score returned by the CRF supertagger, given the input token x. The voting function `Vote` selects a tag predicted by the majority of systems by counting votes. The F1 function `F1` maps the dependency label assigned by the system, which was the most accurate on the development set.

In the second case (54-b), one of the dependency labels predicted by the CG-based tagger was identical with that returned by the CRF-based tagger. In the voting, the tags agreed by the CG-based tagger and the CRF-based tagger were selected if the confident scores were above the threshold of 0.76. Otherwise, one of the dependency grammar labels returned by the CG-based tagger won if the first back-off (ParZu parser) predicted it. Otherwise, the second back-off (Bohnet parser) was introduced to vote in the same way as the first back-off. As the last resort, F1 scores were used to decide the winners.

$$\begin{cases} CRF(x) = CG(x) & \text{if } CRF(x) \cong CG(x) \text{ and } Conf(x) \geq 0.76 \\ Vote(CRF(x), CG(x), PARZU(x)) & \text{else} \\ Vote(CG(x), PARZU(x)) & \text{else} \\ Vote(CRF(x), CG(x), BOHNET(x)) & \text{else} \\ Vote(CG(x), BOHNET(x)) & \text{else} \\ F1(CG(x), CRF(x), PARZU(x), BOHNET(x)) & \text{else} \end{cases} \tag{5.7}$$

In the third case (54-c), the CG-based tagger and the CRF-based tagger did not agree. Then all four committee members voted. In the case that voting tied the score, then the CG-based tagger's vote was weighted (doubled) if the token was a part of complex noun phrases or list elements. These syntactic structures are domain-specific and tuned in the CG-based tagger. If the voting did not result in a decision, then the CG-based tagger

was excluded from the committee. Therefore, the CRF-based tagger, the ParZu and the Bohnet parser voted. As the last resort, F1 scores were used to decide the winners.

$$\begin{cases} Vote(CG(x), CRF(x), PARZU(x), BOHNET(x)) & \text{if } CRF(x) \neg CG(x) \\ Vote(CG(x) * 2, CRF(x), PARZU(x), BOHNET(x)) & \text{if domain-specific structures} \\ Vote(CRF(x), PARZU(x), BOHNET(x)) & \text{else} \\ F1(CG(x), CRF(x), PARZU(x), BOHNET(x)) & \text{else} \end{cases} \tag{5.8}$$

The fourth case (54-d) is the case of prepositions. In this case, the CG-based tagger did not provide a distinction between two dependency labels for prepositions `OBJP` and `PP`. Therefore, the CRF-based tagger, the ParZu parser, and the Bohnet parser gave simple votes.

$$\begin{cases} Vote(CRF(x), PARZU(x), BOHNET(x)) & \text{if } CRF(x) = PP \text{ or } OBJP \end{cases} \tag{5.9}$$

I tested the developed ensemble system on the same data that were used in the development set (Law200Test). The overall label accuracy was 95.32%. The ensemble system outperformed both the CRF-based tagger (92.86%) and the best confidence-based voting system (93.78%).

	Hybrid	
Combined with	CG-based tagger, CRF-based hybrid tagger, ParZu and graph-based Bohnet	
Tested on	Law200Test	
Overall label accuracy	95.32%	(1956, 96)
SUBJ	0.97	(0.97, 0.98)
OBJA	0.93	(0.91, 0.95)
OBJD	0.77	(0.74, 0.80)
OBJG	0.67	(1.00, 0.50)
GMOD	0.96	(0.96, 0.95)
APP	0.82	(0.79, 0.85)
KON	0.95	(1.00, 0.90)
CJ	0.97	(0.96, 0.98)
OBJP	0.61	(0.78, 0.50)
PP	0.97	(0.95, 0.98)

TABLE 5.15: Hybrid Tagging

In conclusion, I integrated the rule-based tagger as morphological constraints into the CRF-based tagger by adding a small amount of in-domain annotated data (300 sentences) to a large amount of out-of-domain training data selected for the domain. The in-domain data were augmented by the case features and dependency grammar relations that were returned by the rule-based supertagger. The feature augmentation helped the CRF-based model to outperform the state-of-the-art parsers in label accuracy. This hybrid CRF-based tagger was then combined again with the rule-based supertagger and with the two state-of-the-art parsers in a voting method. The voting method was based on the confidence score and a back-off voting that was based on the results of the data analysis. This ensemble supertagger outperformed the hybrid CRF-based tagger on the development set. In the next section, I evaluate the performance of the best supertagger — the ensemble tagger — on unknown data.

5.5 Evaluation

In this evaluation, I tested the back-off voting ensemble supertagger presented in the previous section on 200 randomly selected and newly annotated sentences in Swiss German-language law texts (Law200Eval). As discussed in Chapter 4, 200 sentences provide a sufficient amount of to discover the major problems in a system.

Evaluation Results Table 5.16 illustrates the results. Overall, the ensemble tagger outperformed both the rule-based tagger described in Chapter 3 and the hybrid CRF-based tagger described in 5.4.1. The ensemble tagger achieved the best F1 score in all main dependency grammar relations listed in Table 5.16 with two exceptions: dative objects (`OBJD`) and coordination (`KON`).

	CG-based tagger		CRF-based tagger		Ensemble tagger	
Description	Rule-based tagger [without PP and OBJP]		Hybrid: CG and CRF		Hybrid: CG, CRF, ParZu and Bohnet	
Tested on	Law200Eval		Law200Eval		Law200Eval	
Overall label accuracy	83.37%	(1414, 282)	90.16%	(1824, 199)	94.22%	(1906, 117)
SUBJ	0.82	(0.81, 0.84)	0.88	(0.83, 0.94)	**0.93**	(0.92, 0.95)
OBJA	0.65	(0.65, 0.65)	0.82	(0.80, 0.84)	**0.89**	(0.86, 0.91)
OBJD	0.48	(0.88, 0.33)	**0.85**	(0.87, 0.84)	0.81	(0.83, 0.78)
GMOD	0.88	(0.85, 0.91)	**0.96**	(0.98, 0.93)	**0.96**	(0.99, 0.93)
APP	0.63	(0.49, 0.87)	0.65	(0.60, 0.70)	**0.85**	(0.81, 0.90)
KON	**0.88**	(0.87, 0.89)	0.78	(0.89, 0.69)	0.87	(0.89, 0.84)
CJ	0.93	(0.92, 0.95)	0.91	(0.94, 0.88)	**0.95**	(0.95, 0.94)
OBJP	_	_	0.45	(0.39, 0.53)	**0.74**	(0.61, 0.93)
PP	_	_	0.96	(0.96, 0.96)	**0.98**	(0.98, 0.97)

Table 5.16: Evaluation: Hybrid Tagging

In addition, I compared the ensemble tagger with the two parsers that were used for the voting: the graph-based Bohnet parser and the ParZu parser. The ensemble supertagger outperformed these parsers in both label accuracy and dependency relations (cf. Table 5.17).

	ParZu		Bohnet (Graph-based)	
Tested on	Law200Eval		Law200Eval	
Overall label accuracy	88.14%	(1783, 240)	87.54%	(1771, 252)
SUBJ	0.90	(0.87, 0.94)	0.87	(0.85, 0.89)
OBJA	0.82	(0.77, 0.87)	0.77	(0.79, 0.75)
OBJD	0.77	(0.77, 0.77)	0.69	(0.60, 0.82)
GMOD	0.94	(0.95, 0.92)	0.95	(0.97, 0.92)
APP	0.58	(0.52, 0.66)	0.59	(0.48, 0.75)
KON	0.62	(0.50, 0.81)	0.62	(0.50, 0.83)
CJ	0.93	(0.92, 0.95)	0.94	(0.94, 0.94)
OBJP	0.68	(0.65, 0.71)	0.55	(0.57, 0.54)
PP	0.96	(0.94, 0.98)	0.95	(0.93, 0.97)

Table 5.17: Evaluation: ParZu and Bohnet

Error Analysis I then analyzed 100 errors manually. One third of the mistakes (43 tokens) were rooted in morphosyntactic ambiguity. Morphosyntactic ambiguity typically

causes competition between either subject and direct object or dative object and genitive modification. In these cases, the CG-based tagger returned multiple tags. The threshold for the confidence score of the hybrid tagger was set low (0.76), which made the back-off voting system vulnerable and overpowered by the hybrid CRF-based tagger. The example (55) illustrates that the CG-based tagger returned both subject and direct object for the token *Departement*. The CRF-based tagger then returned a wrong tag (here direct object) with a relatively high confidence score (here 0.96). In this case, the ParZu parser and the Bohnet parser both returned subjects. As the results of the experiments showed, the simple voting system achieved the lowest label accuracy. An easy solution to this problem would be to set the threshold higher. However, in several cases, the parsers did not provide better predictions. For example, the CG-based tagger returned two dependency relations: subject (`SUBJ`) and direct object (`OBJA`) for the word *die* in the sentence (56). The CRF-based tagger returned direct object (`OBJA`) with a confidence score of 0.81. Because the confidence score is above the threshold, the ensemble supertagger assigned the dependency relation tag `OBJA`. The threshold may be too high. However, because the word *die* is morphosyntactically highly ambiguous, the parsers analyzed them as determiners because of the POS-taggers in this example.

(55) Widersetzt sich der Empfangsstaat nicht, so kann das Eidgenössische **Departement** für auswärtige Angelegenheiten ausnahmsweise eine Vertretung beauftragen, zugunsten von Staatenlosen oder Flüchtlingen mit gewöhnlichem Aufenthalt in der Schweiz zu intervenieren

(56) Wer Biozidprodukte, **die** pathogene Mikroorganismen sind oder solche enthalten, in Verkehr bringen will, muss die Sicherstellungspflichten nach Artikel 14 FrSV erfüllen.

The second group of the most common errors (32 errors) was related to the strong indicators of local contexts. This type of error is typically caused by heavily weighting the CG-based tagger in the architecture of the ensemble supertagger. The mistakes made by the CG-based tagger are easily propagated into the hybrid CRF-based tagger if the CG-based tagger is generally accurate in the prediction of the targeted dependency tag and if there are good indicators for the CRF. If the CRF-based tagger agrees with the CG-based tagger, it is difficult to correct the agreed tags in the back-off voting method because they are handled as first class committee members. For example, the token *Aufgaben* in the sentence (57) is a typical example in which the mistake made by the CG-based tagger is not corrected by the hybrid CRF-based tagger. The high confidence score prevents the introduction of back-off systems in the ensemble system. The CG-based tagger and the hybrid CRF-based tagger returned the coordination relation `KON`,

the ParZu parser returned direct object (OBJA), and the Bohnet parser did not return anything. The confidence score was high (0.998). The comma was used as a local indicator for the CG-based tagger and the CRF-based tagger. In a similar case, the token *Institutionen* in the sentence (58) was the apposition, and the indicator was the adverb *namentlich*. Because the CG-based tagger does not include much lexical information, contextually available categorical local cues are often used directly as indicators. In this case, the comma was used as the cue for KON in the CG-based tagger. The comma and the adverb were used as features in the CRF-based tagger. The cue word *namentlich* for apposition was not sufficiently strong to overcome the comma feature and the output feature (here KON) returned by the CG-based supertagger. Consequently, the hybrid CRF-based tagger returned the coordination relation KON. In this case, the back-off voting did not help to correct the relation KON into APP.

(57) Das BBT hat im Bereich der internationalen Programme und Projekte, unter dem Vorbehalt von Absatz 2, die folgenden **Aufgaben**:

a. [...]

(58) Die Zusammenarbeit mit den Staaten Osteuropas hat folgende Ziele:

a. Förderung und Stärkung der Rechtsstaatlichkeit und der Menschenrechte sowie Aufbau und Festigung des demokratischen Systems, namentlich stabiler politischer **Institutionen**; [...]

5.6 Conclusion

In this chapter, I presented a method for modeling the encoding of arguments in German. I developed a statistical supertagger based on the sequential conditional random fields (CRF) model. In the CRF model, multivariate linguistic factors of the realization of word order in German were integrated as features, which statistically solved encoding problems caused by free word order. The CRF-based supertagger assigned dependency relations to nouns and prepositions with a high degree of accuracy, which was comparable with that of POS tagging.

To optimize the CRF models for Swiss German-language law texts, the training data were adapted in two ways: 1) selecting in-domain-like training instances from a large amount of out-of-domain training data; and 2) adding a small amount of in-domain data as training data. The small amount of the in-domain data was augmented by the case features and dependency grammar labels returned by the rule-based supertagger. Hence, the hard morphological constraints were integrated into the CRF model, which

drastically improved the score of label accuracy. Morphosyntactic ambiguity caused by case syncretism was thus solved by integrating possible case features directly into the CRF model in parallel with the other linguistic factors.

To boost accuracy further, two state-of-the-art parsers were then combined with the hybrid supertagger and the rule-based tagger by using a back-off voting method. The results showed that a simple voting system did not outperform the standalone tagger. However, the back-off voting method, which was based on the data analysis, achieved a label accuracy of 94.22% on the test set.

5.7 Summary and Overview of the Supertagger Components

In Figure 5.1, I provide a summary and overview of the various components developed and used in the supertagger in my work. The whole supertagger comprises of three components: (1) a constraint-grammar-based supertagger, (2) a hybrid CRF-based supertagger, and (3) an ensemble supertagger. The constraint-grammar-based supertagger analyses topological fields, morphosyntactic features and dependency grammar relations. Because the constraint-grammar-based tagger analysed topological fields on a test set with high F1 scores (cf. Chapter 4), I used it for the final topological field analysis. Since the constraint-grammar-based tagger assigns multiple dependency grammar labels to nouns and prepositions, its outputs are used as features in the hybrid CRF-based supertagger and in the ensemble supertagger. The hybrid CRF-based supertagger is trained on the data set comprising of a small amount of the in-domain data and a large amount of the out-of-domain data (TüBa) adopted to the domain, and uses resources such as GermaNet as features (cf. Section 5.4.1). The supertagger is then combined with the constraint-grammar-based supertagger and two parsers (cf. Section 5.4.2). As the evaluation of this chapter showed that the ensemble supertagger improved over the hybrid CRF-based supertagger in assigning dependency grammar labels, I use the dependency grammar labels returned by the ensemble supertagger as final output.

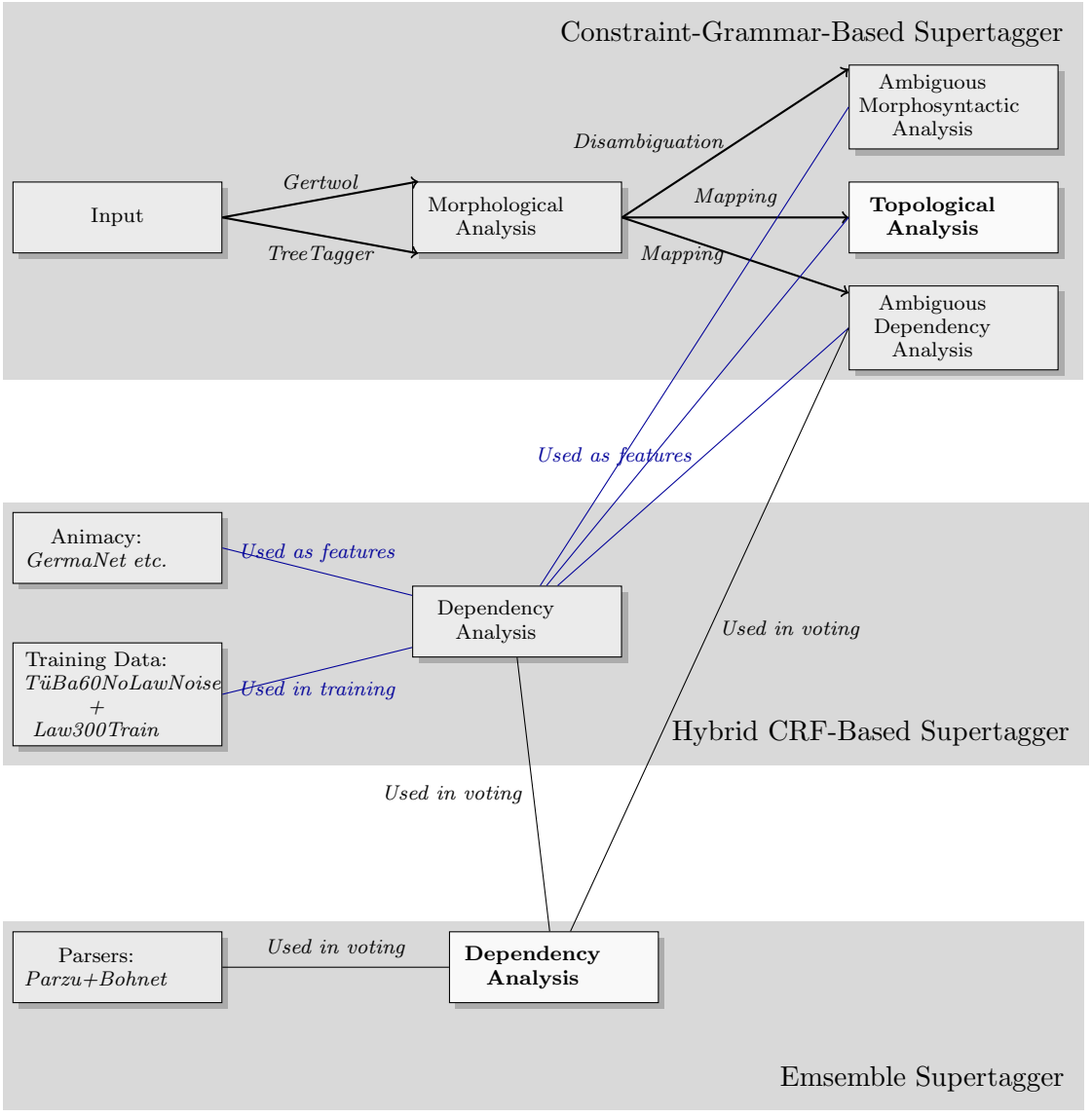


FIGURE 5.1: Overview of three supertaggers

Chapter 6

Evaluation and Improvement

> *Kommen die Forderungen der Handwerker und Unternehmer bei der Pfandverwertung zu Verlust, so ist der Ausfall aus dem den Wert des Bodens übersteigenden Verwertungsanteil der vorgehenden Pfandgläubiger zu ersetzen, sofern das Grundstück durch ihre Pfandrechte in einer für sie erkennbaren Weise zum Nachteil der Handwerker und Unternehmer belastet worden ist.*
>
> – Swiss Civil Code Art. 841 para. 1

In this chapter, I test the error detection method that I presented in Chapter 2 by using the newly developed supertagger (cf. Chapters 3, 4, and 5). Methodologically, the error detection presented in Chapter 2 was rule-based and comprised a set of problematic syntactic structure types to anticipate violations of style guidelines. The error detection was context-independent because every syntactic structure that matched the error types was regarded as stylistically incorrect. In this chapter, I introduce a context-dependent error detection method that differentiates individual violations according to context and compare it with the context-independent approach.

6.1 Context-Independent Error Detection: Use of the Supertagger

In Chapters 3, 4, and 5, I developed a domain-specific supertagger to improve the accuracy of the error detection method presented in Chapter 2. In Chapter 2, a state-of-the-art statistical dependency grammar parser and a topological dependency grammar parser were used for the syntactic annotation. In this section, I test the same error

detection, but I use the supertagger as the pre-processing syntactic analyzer. I use the same test data and same evaluation setting as in Chapter 2.

Test Table 6.1 shows the results. By using the supertagger, the automatic detection achieved an F1 score of 83.78% and outperformed the error detection that used the dependency parser and the topological dependency parser presented in Chapter 2 (F1 scores of 64.26% and 75.67%, respectively). Table 6.2 summarizes the evaluation results of the style error detection developed in Chapter 2, using the dependency parser and the topological dependency parser. The error detection using the supertagger outperformed that with the dependency parser and topological dependency parser for 6 of 7 style rules according to F1 scores and 5 of 7 style rules in precision and recall.

(#) Target	Precision	(retrieved #)	Recall	(total #)	F1
(1A) Agentive passive sentences	92.50	(40)	74.00	(50)	82.22
(1B) Marked word orders	**100.00**	(17)	**34.00**	(50)	**50.75**
(1C) Coordinated sentences	**100.00**	(45)	**90.00**	(50)	**94.74**
(1D) Embedded clauses	**96.55**	(29)	**56.00**	(50)	**70.89**
(1E) Complex noun phrases	**97.87**	(47)	92.00	(50)	**94.84**
(2A) Consequences before conditions	**90.57**	(53)	**96.00**	(50)	**93.21**
(2B) Separated conditions	83.33	(54)	**90.00**	(50)	**86.54**
Total	**93.33**	(285)	**76.00**	(350)	**83.78**

TABLE 6.1: Evaluation of the error detection system using the topological dependency supertagger

(#) Target	Precision		Recall		F1	
	Dep.Par	Top.Par	Dep.Par	Top.Par	Dep.Par	Top.Par
(1A) Agentive passive sentences	**93.48**	89.74	**86.00**	70.00	**89.58**	78.65
(1B) Marked word orders	100.00	100.00	18.00	20.00	30.51	33.33
(1C) Coordinated sentences	84.09	83.67	74.00	82.00	78.72	82.83
(1D) Embedded clauses	77.78	69.70	14.00	46.00	23.73	55.42
(1E) Complex noun phrases	97.67	85.45	84.00	**94.00**	90.32	89.52
(2A) Consequences before conditions	58.33	88.46	14.00	92.00	22.58	90.20
(2B) Separated conditions	84.21	**87.80**	64.00	72.00	72.73	79.12
Total	88.06	85.30	50.57	68.00	64.26	75.67

TABLE 6.2: Evaluation of the error detection system using the dependency parser (Dep.Par) and topological dependency parser (Top.Par): The data is derived from Table 2.5 and 2.12 in Chapter 2

Error Analysis To gain insight into the types of errors made by the detection system, I conducted an error analysis of false positives. Table 6.3 shows the results. The number of errors caused by the preprocessing of syntactic annotation was drastically reduced compared to the dependency grammar parser (14 errors) and the topological dependency

grammar parser (27 errors)(cf. Table 2.5 and 2.12 in Chapter 2). The detection errors were mostly the results of insufficient recognition of discourse relation (condition) (cf. (2A) and (2B) in Table 6.3).[1]

(#) Target	Preprocessing errors	Parsing errors	Detection errors
(1A) Agentive passive sentences	1	1	1
(1B) Marked word orders	0	0	0
(1C) Coordinated sentences	0	0	0
(1D) Embedded clauses	0	1	0
(1E) Complex noun phrases	0	1	0
(2A) Consequences before conditions	0	0	5
(2B) Separated conditions	0	0	15
Total	1	3	21

TABLE 6.3: Error analysis of false positives: error detection system using the topological dependency supertagger

To provide an overview of the number of style violations that could be found in the actual law texts, I ran the error detection system with the supertagger on the entire collection of Swiss German-language law texts (cf. Section 1.5). Table 6.4 provides an overview of the number of retrieved style violations. The table shows that violations of the rules *complex noun phrases* ((1E) in Table 6.3) and *consequences before conditions* ((2A) in Table 6.3) were far more frequent than the violations of other style rules were. As the detection of these rules was high in precision (97.87% and 90.57%) and in recall (92%, 96%) (cf. Table 6.1), the numbers of detected instances roughly correspond to the ones of the style errors contained in the Swiss German-language law texts. In comparison, the retrieved numbers of the style rules (1C) and (2B), of which the quality of the error detection was comparative (cf. Table 6.1), were far lower (cf. Table 6.4).

[1] As discussed in Section 2.4.3, the (2A) and (2B) rules are based on the recognition of conditions, which failed mostly because of the ambiguity of the preposition *bei*.

(#) Target	Total retrieved
(1A) Agentive passive sentences	7,035
(1B) Marked word orders	6,365
(1C) Coordinated sentences	7,054
(1D) Embedded clauses	8,032
(1E) Complex noun phrases	24,777
(2A) Consequences before conditions	19,574
(2B) Separated conditions	3,522
Total	76,359

TABLE 6.4: The number of syntax-related style violations in Swiss German-language law texts

In conclusion, this section presented an evaluation of the error detection system that used the newly developed topological dependency supertagger. By using the supertagger, the automatic detection achieved an F1 score of 83.78% and outperformed the error detection using the dependency parser and the topological dependency parser presented in Chapter 2. Furthermore, the results showed that the rule-based error detection system achieved high accuracy without annotating a large amount of in-domain data. In the development, I used a small amount of annotated data (cf. Section 2.3).

In addition, the evaluation showed that some style rules were far more frequently violated than other rules were. In the next section, I present a method to filter out some of these detected structures, so that the error detection system does not fire too many warnings. For the filtering, I focused on the style rule 'complex noun phrases' because it was retrieved the most frequently from the actual law texts (cf. Table 6.4).

6.2 Context-Dependent Error Detection

6.2.1 Introduction

Despite dissuasion in style guidelines, complex noun phrases are used frequently in actual law texts. This indicates that complex noun phrases are accepted by legislative editors to some extent. The general rule in the guidelines for rephrasing complex noun phrases as relative clauses is only loosely adhered to in practice, and violations of this rule can be found 10,000 fold (cf. Table 6.4).

The problem might be because the style rules in guidelines are not specific enough. They are too simplistic to be applied in an automatic detection of the violations. A rule-based system requires style rules to be so concrete that the detection of the violations can be operationalized for machines (cf. Section 2.2.1). Even if the examples in the guidelines

show stylistically incorrect complex noun phrases (1-a) (cf. Section 2.2), it is difficult to decide a clear-cut threshold for style violations. We do not know, for example, whether simplified participle phrases (1-b-i) and (1-b-ii) are both acceptable in the domain or if only one of them is acceptable. Both could be unacceptable.

(1) a. Bad example in the guideline of Zurich:
Grund *der am 1. Januar des dem Auszahlungsjahr vorangehenden Jahres im Kanton bekannten definitiven Steuerfaktoren* (Regierungsrat des Kantons Zürich, 2005, pp. 74)

b. Simplified versions of (1-a)

(i) Grund *der am 1. Januar des dem Auszahlungsjahr vorangehenden Jahres bekannten definitiven Steuerfaktoren* (self-rephrased)

(ii) Grund *der im Kanton bekannten definitiven Steuerfaktoren* (self-rephrased)

A rule-based detection of style guideline violations is limited in modeling the grade of style errors because the latter 'depend on the writer, the reading audience, and the type of document involved' (Ravin, 1988, pp. 109). Moreover, '(the) severity depends on quantitative measures' (ibd.). To overcome this limitation, Ravin (1988) left the task to the user, who could decide the threshold. His IBM CRITIQUE was not strictly domain-specific and was used broadly within IBM, the government, and schools. Because our system is domain-specific, I will develop a method to model the use of complex noun phrases which is conventionalized in the domain. The goal is to differentiate individual complex noun phrases depending on the grade of error.

To understand the tipping point between acceptable and unacceptable syntactic complexity of complex noun phrases and to study the writing style in the domain empirically, I employ statistical classification as a method. I define writing style as the choice between alternatives following the stylistic definition (Eroms, 2008, pp. 5ff). Here, the syntactic choice to be investigated is the one between complex noun phrases and relative clauses. They can mutually be alternated without loosing much semantic content. To model the syntactic choice, I build a classifier that reflects the actual use of complex noun phrases and relative clauses in current Swiss German-language law texts as evidence. I hypothesize that (i) the main factor in decisions about syntactic choice is syntactic complexity and (ii) complex noun phrases that are similar to relative clauses in syntactic complexity are more likely to be reformulated by legislative editors. To operationalize the modeling of syntactic choice, decision factors, i.e. syntactic complexity grades, are integrated as features. The classifier then learns how these two syntactic alternations were actually selected in the current Swiss German-language law texts based on these decision factors. The task of the classifier is to discriminate complex noun phrases into

two classes based on the syntactic complexity: (i) stylistically adequate complex noun phrases and (ii) stylistically inadequate complex noun phrases (i.e. relative-clause-like complex noun phrases). The complex noun phrases classified as adequate are filtered out and are ignored in the error detection.

The following subsections are organized as follows: I begin by choosing the statistical methods and data used in the classification task, and I then determine the decision factors of the syntactic choice, i.e. syntactic complexity, in detail. I then test the methods and features and evaluate the learned statistical models on a small number of sentences, which will be then stylistically judged by a legislative editor.

6.2.2 Method, Feature and Data

Methods For the classification of stylistically adequate or inadequate complex noun phrases, I use supervised machine learning methods, particularly Naive Bayes and Support Vector Machines (SVM). Naive Bayes (NB) is a simple and efficient supervised method based on the naïve assumption of Bayes' theorem that features are mutually independent given the context of the class. I use two types of NB: Gaussian NB and multinominal NB. In Gaussian NB, the likelihood of the features is assumed to be Gaussian, that is, normally distributed, whereas it is assumed to be multinominal in multinominal NB. SVM is a hyperplane-based discriminative classifier and is the state-of-the-art method for classification. In corpus linguistics, logistic regression has been widely used to investigate multifactorial language data in syntactic alternation, particularly dative shift (Bresnan et al., 2007, Arnold et al., 2000), genitive alternation (Gries and Wulff, 2013), and heavy NP shift (Arnold et al., 2000). Compared with other machine learning methods, logistic regression is advantageous in linguistic research because the estimated coefficients (or parameters) are explanatory and help in interpreting how they affect the probabilities of events (cf. Baayen, 2011). However, I use these three machine-learning methods because of their high performance in NLP tasks. NB methods have been successfully applied to text documentation (cf. McCallum and Nigam, 2001). SVM has been successfully applied in NLP tasks, such as in parsing (e.g. Nivre et al., 2006) and text documentation (e.g. Joachims, 1998).

Feature Syntactic complexity that affects syntactic choice might be the following:

(2) (A) A complex noun phrase,
(B) The maximally projected noun phrase of a complex noun phrase
(C) The sub-fields of a complex noun phrase (i.e. vorfeld, mittelfeld, nachfeld).

For the measurement of the syntactic complexity in each syntactic context (2), the number of occurrences of syntactic categories is measured by following the methods of readability assessment. In recent readability assessments, classifiers are used to judge the readability of a document and a sentence (e.g. linear regression (Feng et al., 2009), multiple regression (Chen and Zechner, 2011), decision tree (Chen and Zechner, 2011), support vector machine (Falkenjack and Jönsson, 2014, Dell'Orletta et al., 2011, Nenkova et al., 2010, Petersen and Ostendorf, 2009, Pitler and Nenkova, 2008, Schwarm and Ostendorf, 2005). In statistical models, linguistically and cognitively motivated features are integrated as indicators of the comprehensibility of a text and a sentence. In most readability assessment systems, syntactic complexity is represented as features by counting the occurrences of some syntactic structures, such as noun phrases or prepositional phrases (cf. Table 6.5).

Features	Language, Target readers and References
The average number of adjectives per sentence	English, Adults with intellectual disabilities (Feng et al., 2010, 2009)*
The average number of common and proper names per sentence	English, Adult with intellectual disabilities (Feng et al., 2010, 2009)*
The average number of noun phrases per sentence	English, L2 learners (Petersen and Ostendorf, 2009, Schwarm and Ostendorf, 2005); English, Adult with intellectual disabilities (Feng et al., 2010, 2009)*; Spoken English, age-specific (Vajjala and Meurers, 2014); English, general (Kate et al., 2010)
The average number of verb phrases per sentence	L2 learners (Petersen and Ostendorf, 2009, Schwarm and Ostendorf, 2005); English, Adult with intellectual disabilities (Feng et al., 2010, 2009)*; English, general (Kate et al., 2010)
The average number of verbless phrases per sentence	English, general (Kate et al., 2010)
The average number of prepositions per sentence	Spoken English, age-specific (Vajjala and Meurers, 2014)
The average number of prepositional phrases per sentence	English, Adult with intellectual disabilities (Feng et al., 2010, 2009)*; Spoken English, age-specific (Vajjala and Meurers, 2014); English, general (Kate et al., 2010)
The average number of all types of phrases per sentence	English, general (Kate et al., 2010)
The average number of all types of phrases per clause	English, general (Kate et al., 2010)
The average number of SBARs (i.e. subordinated clauses) per sentence	English, L2 learners (Petersen and Ostendorf, 2009, Schwarm and Ostendorf, 2005); English, Adult with intellectual disabilities (Feng et al., 2010, 2009)*
The average number of clauses per sentence	Spoken English, age-specific (Vajjala and Meurers, 2014)
Average parse tree height	English, L2 learners (Petersen and Ostendorf, 2009, Schwarm and Ostendorf, 2005); Spoken English, age-specific (Vajjala and Meurers, 2014)
Depth of clause center embedding	German, General (vor der Brück et al., 2008)
Distance between verb and separable prefix	German, General (vor der Brück et al., 2008)
Number of words per maximal projected noun phrases	German, General (vor der Brück et al., 2008)

TABLE 6.5: Syntactic complexity measurement in readability assessment (the work marked by star * includes also the total number of the specific syntactic structures in documents)

To operationalize the syntactic choice, I created the syntactic features so that the features of relative clauses (RC) correspond to those of complex noun phrases (complex NP). The correspondence is based on the operation of rephrasing from relative clauses to complex noun phrases. For the rephrasing, three operations are conducted, as illustrated in (3). In the example of the alternation pair (4), the relative clause can be rephrased into a complex noun phrase by removing the relative pronoun *das* (cf. the first operation (3-a)), by converting the predicate *eingesetzt* into the attributive adjective *eingesetzte*

(cf. the second operation (3-b)), and by removing the auxiliary verb *wird* (cf. the third operation (3-c)).

(3) a. Relative pronouns are removed.
b. Predicative verbs are converted into attributive participles, zu-infinitives and adjectives.
c. Auxiliary verbs such as *werden, sein, haben* and modal verbs are removed.

(4) a. **RC**
Das Personal, *das nach diesem Gesetz zum Schutz von Personen, Behörden und Gebäuden eingesetzt wird*, darf zur Erfüllung seines Auftrags und, soweit die zu schützenden Rechtsgüter es rechtfertigen, polizeilichen Zwang und polizeiliche Massnahmen anwenden. (self-rephrased from (4-b))

b. **Complex NP**
Das nach diesem Gesetz zum Schutz von Personen, Behörden und Gebäuden eingesetzte Personal darf zur Erfüllung seines Auftrags und, soweit die zu schützenden Rechtsgüter es rechtfertigen, polizeilichen Zwang und polizeiliche Massnahmen anwenden.[2]

Table 6.6 provides an overview of the syntactic features in our task. These 14 features are the number of syntactic categories that belong to one of three syntactic contexts (A), (B), and (C) (cf. (2)). Each feature in each class corresponds to others in the alternation. I illustrate the features using the sentence pair (5-a) and (5-b).

[2] All examples in this chapter are derived from Swiss German-language law texts (cf. Section 1.5)

	Features	Corresponding tokens in (5)	#
(A1)	**Complex NP:** # of tokens in the embedded phrase of complex NP **RC:** # of corresponding tokens in RC	*für die Erfüllung ... notwendigen/notwendig*	9
(A2)	# of nouns and pronouns in (A1)	*Erfüllung, Schutzpflichten, Schweiz*	3
(A3)	# of prepositions, comparative conjunctions and pronominal adverbs in (A1)	*für*	1
(A4)	# of commas and coordinating conjunctions in (A1)	_	0
(A5)	# of predicates in (A1)	*notwendigen/notwendig*	1
(A6)	**Complex NP:** # of tokens in the head NP of a complex NP **RC:** # of tokens in an antecedent NP	*die, Massnahmen*	2
(B1)	# of remaining tokens in the projected NP of (A6)	*auf ihrem Gebiet*	3
(B2)	# of nouns and pronouns in (B1)	*Gebiet*	1
(B3)	# of prepositions, comparative conjunctions and pronominal adverbs in (B1)	_	0
(B4)	# of commas and coordinating conjunctions in (B1)	_	0
(C1)	# of remaining tokens in the sub-field of (B1)	*in Absprache mit fedpol*	4
(C2)	# of nouns and pronouns in (C1)	*Absprache, fedpol*	2
(C3)	# of prepositions, comparative conjunctions and pronominal adverbs in (C1)	*in, mit*	2
(C4)	# of commas and coordinating conjunctions in (C1)	_	0

TABLE 6.6: Features for the classification of complex noun phrases and relative clauses

(5) Feature A = *italic parts*, Feature B = **bold parts**, Feature C = underlined parts

a. Die Kantone treffen in Absprache mit fedpol *die für die Erfüllung der völkerrechtlichen Schutzpflichten der Schweiz notwendigen Massnahmen* **auf ihrem Gebiet**; ... (self-rephrased)

b. Die Kantone treffen in Absprache mit fedpol *die Massnahmen* **auf ihrem Gebiet**, die *für die Erfüllung der völkerrechtlichen Schutzpflichten der Schweiz notwendig* sind; ...

Data For the training data and cross-validation data, I used current law texts as evidence and gold data in classification. The automatically recognized complex noun phrases and relative clauses are regarded as instances of the classes *complex noun phrase* (stylistically correct complex noun phrases) and *relative clause* (stylistically incorrect complex noun phrases), respectively. I assume that the majority of complex noun phrases and relative clauses in current law texts are stylistically correct, and the minority of those syntactic structures would be better rephrased as relative clauses. Swiss German-language law texts are edited by linguistic experts, which supports my assumption.

To create the training and cross-validation sets, I extracted complex noun phrases and

relative clauses from Swiss German-language law texts (cf. Section 1.5). For the extraction, the texts are parsed with the supertagger and are processed with the rule-based error detection system described in Chapter 2.[3] Next, I automatically extract the features from the extracted complex noun phrases and relative clauses. To identify antecedents for relative pronouns, I use the German co-reference resolution system, CorZu (Klenner and Tuggener, 2011).[4] For the recognition of maximally projected noun phrases, I use an NP chunker (cf. Appendix E).

The complex noun phrases and relative clauses augmented by features are then divided into a training set, a cross-validation set, and an evaluation set. For the evaluation, two law texts were selected from different periods because writing style evolves over time: One text is the civil code. Written in 1907, it is one of the oldest law texts in Switzerland. The second text is the animal protection law, which can be considered a current law text that was written in 2005. The remaining items of extracted complex noun phrases and relative clauses are separated into the training set (80%, each 17,420 items for complex noun phrases and relative clauses) and the cross-validation set (20%, each have 4,355 items for complex noun phrases and relative clauses). In this work, cross-validation is used to choose the best classifiers and features for the task.

6.2.3 Experiments

To decide which supervised machine learning methods and which feature types are ideal for the classification task, I trained all three classifiers,[5] Gaussian Naive Bayes (GNB), Multinominal Naive Bayes (MNB) and Support Vector Machines (SVM), with three types of features: syntactic complexity of the local context (A), medium context (A) and (B) and global context (A), (B) and (C) (cf. (A), (B), (C) in Table 6.6).

In the experiment, I tested the trained models on the cross-validation data of complex noun phrases and relative clauses. I evaluated the classification for each class and computed the F1 score, the precision, and the recall. The results are shown in Table 6.7. All three features of the SVM model achieved the best F1 score, precision, and recall in the prediction of class *complex noun phrase* and F1 score and precision in the prediction of class *relative clause*. The Gaussian NB Model with features A and B outperformed

[3] All extracted relative clauses are used with the exception of relative clauses in which the relative pronoun is dependent on a preposition.

[4] CorZu is available at: `http://www.cl.uzh.ch/research/coreferenceresolution.html` (last visited on 01/06/2015)

[5] I used the implementation of sci-kit learn: `http://scikit-learn.org/stable/` (last visited on 01/06/2015)

the SVM model in the score for recall in the prediction of class *relative clause*. The SVM model constantly increased the F1 score, precision, and recall by increasing the number of features. The Gaussian and multinomial NB models also tended to do the same, which indicates that not only the local contexts but also wider contexts affected the syntactic choice.

	Local context Feature A		*Medium context* Feature B + C		*Global context* Feature A + B + C	
	Complex NP	RC	Complex NP	RC	Complex NP	RC
GNB	71.80	53.02	68.01	77.53	80.08	81.64
	(59.84/89.74)	(79.49/39.77)	(86.33/56.10)	(67.48/**91.11**)	(83.65/76.81)	(78.56/84.98)
MNB	74.61	66.51	67.03	75.68	68.61	77.08
	(66.56/84.87)	(79.13/57.36)	(81.52/56.92)	(66.91/87.10)	(84.15/57.91)	(67.92/89.09)
SVM	84.82	83.85	87.13	86.57	**88.84**	**88.49**
	(82.34/87.46)	(86.63/81.24)	(85.35/88.98)	(88.49/84.73)	(**87.52**/**90.20**)	(**89.89**/87.14)

TABLE 6.7: F1 score (precision/recall) of the prediction of class *complex noun phrase* and *relative clause*

Because the objective of the error detection task is to detect relative-clause-like complex noun phrases, the classifier is required to be optimized for the class *relative clause*. I assumed that there are fewer stylistically inadequate complex noun phrases than stylistically adequate ones. To cope with this problem of class imbalance (Longadge and Dongre, 2013), the class *relative clause* is weighted so that the classifier is biased toward the minority class. For class weighting, I used the best model, that is, the SVM model with all three features. The results are shown in Table 6.8. The weighting of the class improved the score of recall with an increase in the weight and a decrease in precision and F1 score. In the class of *complex noun phrases*, the class weighting caused the opposite effects. With the increase in class weighting, the score of precision increased and the score of recall decreased.

	Label accuracy	Complex NP	RC
SVM weight 60	88.27	88.20 (**88.73**, 87.67)	88.33 (87.81, **88.86**)
SVM weight 70	87.11	86.44 (**91.16**, 82.18)	87.71 (83.78, **92.03**)
SVM weight 80	85.68	84.40 (**92.72**, 77.45)	86.77 (80.64, **93.92**)
SVM weight 90	79.89	75.92 (**94.59**, 63.40)	82.73 (72.47, **96.37**)
SVM weight 95	75.27	68.06 (**96.07**, 52.70)	79.82 (67.41, **97.84**)

TABLE 6.8: Experiments on the weighting of class *relative clause* for SVM with global feature (A + B + C)

In conclusion, the best configuration for the classification task is SVM with global feature (A + B + C) in F1 and precision. Adding class weight made the model the best in recall for the minor class *relative clause*.

6.2.4 Evaluation

In this evaluation, I test two hypotheses on the test set: 1) the stylistic choice is affected by syntactic complexity in contexts; and 2) complex noun phrases that are similar to relative clauses with regard to syntactic complexity are more likely to be rephrased into relative clauses by editors.

To examine the first hypothesis, I tested the SVM classifier (global feature A+B+C and no weight) on 269 complex noun phrases in the two law texts selected for the evaluation (cf. Section 6.2.2). As expected, the distinction between two syntactic alternatives was highly accurate. The label accuracy of the classification was 90.33%, that is, 243 instances of 269 were classified as *complex noun phrase*. This result supports my first hypothesis that the syntactic choice between complex noun phrases and relative clauses is affected by syntactic complexity in contexts.[6]

To test the second hypothesis, the classifiers were tested on complex noun phrases that were judged by a legislative editor. In the manual annotation, 50 complex noun phrases were randomly selected from the two law texts for evaluation and given to the legislative editor. He was asked whether he would rephrase these complex noun phrases. Twenty of the 50 phrases were qualified for rephrasing, and 20 were left as they were. In addition, in 10 cases, the editor was not able to make a clear decision for or against re-phrasing. He was also asked to comment freely.

Table 6.9 shows the results of the class *relative clause* prediction for complex noun phrases in precision, recall, and F1 score. The best precision score was 75%, and 6 of 8 items were correctly classified as style violations by the SVM classifier without weighting. The best score for recall was 60%, and 12 out of 20 instances were correctly classified as class *relative clause*, that is, style violations classified by the SVM with a class weight of 90% biased towards class *relative clause*. As expected, weighting the class *relative clause* retrieved more *relative clause* class instances and improved the recall score at the cost of precision. The Gaussian NB with features A and B achieved a better score in recall than the SVM without weight. Although the model is simple, the performance of the Gaussian NB was surprisingly high. However, in total, the classifiers were not high in F1 scores, which indicates that the syntactic choice did not correspond to the criteria that a legislative editor used to rephrase complex noun phrases.

[6] In this work, the number of part-of-speech tags is defined as syntactic complexity. It can be interpreted as the semantic content as well.

Classifier	tp, fp, fn	F1	(precision, recall)
SVM feature A+B+C No Weight	6, 2, 14	42.86	(**75.00**, 30.00)
SVM feature A+B+C weight 60	7, 3, 13	46.67	(70.00, 35.00)
SVM feature A+B+C weight 70	8, 5, 12	48.49	(61.54, 40.00)
SVM feature A+B+C weight 80	9, 7, 11	50.00	(56.25, 45.00)
SVM feature A+B+C weight 90	12, 9, 8	**58.54**	(57.14, **60.00**)
SVM feature A+B+C weight 95	14, 13, 6	42.43	(53.85, 35.00)
GNB Feature A+B	10, 9, 10	51.28	(52.63, 50.00)

TABLE 6.9: Evaluation of the ML-based error detection system (SVM and Gaussian naive Bayes): Prediction of class *relative clause* for complex noun phrases (tp = true positive, fp = false positive, fn = false negative)

For example, the SVM with features A, B and C with class weight 90 that performed at best in the evaluation and optimized for the class relative clauses, predicted the complex noun phrase in the sentence (6) correctly as relative clause. For this complex noun phrase, the editor commented that the complex noun phrase was awful and must be rephrased. For the complex noun phrase in the sentence (7), however, the editor judged it to be rephrased, but noted that [he] "found it not so bad, but it should still rather be rephrased into a relative clause." The SVM erroneously classified it as complex noun phrase. It was clearly difficult for the classifier to correctly predict if the editor judged with weak confidence.

(6) Bei der Grundbuchführung mittels Informatik kommen die Rechtswirkungen *den im System ordnungsgemäss gespeicherten und auf den Geräten des Grundbuchamtes durch technische Hilfsmittel in Schrift und Zahlen lesbaren oder in Plänen dargestellten Daten* zu.

(7) Ist zu befürchten, dass die Erträge oder *die für den Verbrauch bestimmten oder freigegebenen Beträge des Kindesvermögens* nicht bestimmungsgemäss verwendet werden, so kann die Vormundschaftsbehörde auch deren Verwaltung einem Beistand übertragen.

In conclusion, this study supported the first hypothesis that the syntactic choice of complex noun phrases and relative clauses is affected by syntactic complexity in wide contexts. However, the results did not support the second hypothesis that complex noun phrases that are similar to relative clauses with regard to syntactic complexity correspond to style violations and are better rephrased in relative clauses from a legislative editor's point of view.

6.3 Conclusion

In this chapter, I tested the rule-based error detection developed in Chapter 2 by using the supertagger presented in Chapters 3, 4, and 5. The evaluation found that the use of the supertagger outperformed the use of the dependency grammar parser and the topological dependency parser.

In addition, I presented a statistical style-error detection method that took into account the degree of style errors. I built a statistical classifier for the style rule that was retrieved the most often - 'avoidance of complex noun phrases'. I investigated the computational model of stylistic choice between syntactic alternatives (complex noun phrases and relative clauses). The model is based on current Swiss German-language law texts to capture the conventionalized choice of these syntactic alternatives in the domain. I hypothesized that (i) the stylistic choice is affected by syntactic complexity in contexts and (ii) complex noun phrases that are similar to relative clauses with regard to syntactic complexity are more likely to be rephrased into relative clauses by editors. For the automatic classification of two classes, *complex noun phrase* and *relative clause*, I used machine learning methods, naive Bayes and the Support Vector Machine (SVM). For the training of the models, I used instances and features that are automatically extracted from actual law texts under the assumption that the majority of syntactic choices in professionally edited law texts are stylistically correct. The SVM classifier accurately distinguished these two syntactic alternatives based on contextual syntactic complexity. This finding indicates that syntactic choice is affected by syntactic complexity in the context. However, the evaluation also showed that the syntactic choice between alternatives modeled by contextual syntactic complexity did not correspond entirely to the criteria that a legislative editor used to rephrase complex noun phrases.

Chapter 7

Conclusion

7.1 Summary

In this thesis, I presented methods for the automatic annotation and assessment of syntactic structures in Swiss German-language law texts. In this chapter, I summarize the main results of the following: 1) syntactic annotation and 2) syntactic assessment.

Automatic syntactic annotation To automatically assign topological dependency relations to each token and to solve morphosyntactic ambiguities and freedom of word order in German, and to address the domain-specific morphosyntactic distribution and the syntactic complexity, I built a supertagger that combines a rule-based method with a statistical tagging method.

- To achieve high accuracy in the automatic syntactic annotation for an application, I proposed a set of syntactic categories that embody the topological field model and functional dependency grammar. In this tag set, syntactic structures are represented both linearly and functionally with clause boundaries.
- I introduced a rule-based tagging method to model topological fields. The rule-based method was selected because topological fields are a linguistic model describing the rigid linear order of German clause brackets, the relations of which are widely distanced. The tagger contains a set of linear grammar rules that target morphosyntactic features. As grammar rules, I used heuristics that are derived from a linguistic theory (i.e. topological field model), as well as domain-specific preferences. The results of the evaluation showed that the rule-based tagger outperformed both a bigram HMM-based tagger and a state-of-the-art statistical parser tested on Swiss German-language law texts. The results also showed that unlike statistical methods, the performance was not strongly affected by distance.

- To analyze dependency grammar relations automatically, I developed a rule-based tagger that disambiguates morphosyntactic features according to the rules of morphological agreement, argument structures, and psycholinguistic and domain-specific preferences. The case ambiguities of nouns and pronouns were not able to be fully resolved by this rule-based tagger, which led to problems in analyzing the syntactic functions of arguments. Therefore, I then introduced a statistical tagging method because the linear order of arguments is relatively freely permuted, and it is a matter of markedness, that is, frequency. Linguistic factors affecting word order (e.g. animacy and definiteness) are integrated into statistical tagging models such that they interact directly with one another. The accuracy achieved was competitive with that obtained by the state-of-art parsers.

- To integrate linguistic hard constraints into the statistical model, I added a small amount of in-domain training data, which was augmented by the case features and dependency grammar labels obtained by the rule-based supertagger. I then trained a conditional random field (CRF)-based model. The evaluation found that this feature-based hybrid supertagger outperformed three state-of-the-art parsers in label accuracy. This result confirms that the linguistically motivated architecture and features of the supertagger achieved a better performance on domain-specific data even with the use of a small amount of manually annotated training data.

- The feature-based hybrid supertagger was combined again with the rule-based tagger and state-of-the-art parsers through back-off voting in order to boost its accuracy for application (here, for style error detection). The voting method is based on the data analysis of various syntactic analysers' agreement and correlations in performance. The voting-based ensemble supertagger achieved a higher accuracy than the standalone feature-based hybrid supertagger. Its accuracy was comparable with the F1 score of the best part-of-speech (POS) tagging in German texts.

Automatic syntactic assessment To detect violations of style guidelines automatically in the legislative domain, I developed a hybrid error-detection system that comprises rule-based and statistical components. This rule-based system automatically identifies violations of syntax-related style rules based on the anticipation of problematic types of syntactic structures. In addition, the statistical system was able to judge the detected style violations by taking the context into account.

- In order to identify style violations in law texts, I proposed a rule-based error detection system composed of a set of problematic types of syntactic structures based on domain-specific style guidelines. A small development set of stylistically

annotated syntactic structures was sufficient to achieve high accuracy in the detection task. The error analysis showed that this system struggled with mainly the quality of automatic syntactic annotations. In addition, I experimented using preprocessing parsers and investigated the effects of the domain difference between law texts and newspapers with respect to the accuracy of parsing and error detection. The results of the evaluation showed that the domain difference in parsing was evident in the dependency relations at middle and wide distances (i.e. topological fields and syntactic functions of arguments). The results also showed that a high performance of syntactic annotation was a prerequisite for the quality of syntactic assessment. The rule-based syntactic assessment achieved high accuracy because of the domain-specific hybrid ensemble supertagger that was developed in this PhD thesis.

- To filter out some of complex noun phrases that were detected in excess, I proposed a statistical method for automatically categorizing individual complex noun phrases into stylistically adequate complex noun phrases or inadequate complex noun phrases. Stylistically inadequate complex noun phrases are those that are similar to relative clauses with regard to the syntactic complexity. For the automated judgment, I built a classifier that distinguishes complex noun phrases from relative clauses in terms of the syntactic complexity in the context. The classifier was trained on the automatically annotated in-domain texts. Thus, the syntactic choice was modeled on the actual use of syntactic structures conventionalized in the domain. The classifier, a support vector machine (SVM), accurately discriminated one from the other. This result indicates that contextual syntactic complexity is a contributing factor in the syntactic choice between complex noun phrases and relative clauses in the domain. However, the evaluation showed that the modeling of syntactic choice did not entirely correspond to editorial judgment.

In summary, my thesis demonstrated that hybrid automatic syntactic annotation and assessment systems achieved a high accuracy, even in the absence of a large amount of linguistically annotated in-domain data. The experimental results presented in this thesis showed that the use of linguistic theories, data analysis and hybrid methods contributes to the high quality of syntactic annotation and assessment even under the constraint of sparse linguistic resources. My dissertation successfully answered the two research questions, posed in Section 1.3: 1) *How can a high degree of accuracy be achieved in automatic syntactic annotation in domain-specific texts, such as law texts, with sparse in-domain resources for an application?*; and 2) *How can we model domain-specific style error detection without annotated data?*

7.2 Outlook

In the following paragraph, I outline future research that would be interesting to build upon the findings of this thesis.

Automatic syntactic annotation

- **Domain adaptation**
  In this thesis, the proposed method for syntactic annotation was optimized for a specific text type (Swiss German-language law texts) in order to study a new domain that lacked in-domain resources. In the future, the use of NLP methods will spread across more and more applications. In order to be successfully applied in various domains, where there are no or few domain resources available, the question how to effectively and efficiently apply NLP becomes increasingly important.

- **Linguistics in NLP**
  Although I proposed a supertagger for German and Swiss law texts, the methods developed in this thesis can be seen as prototype for other free word order languages and for the integration of markedness in linguistic typology into statistical models. I demonstrated that linguistic and data analysis improve the labeling accuracy above an F1 score of 0.9 despite the fact that in-domain language resources were not available. In the future, linguistics will remain important in NLP research and critical domain applications, regardless of the recent evolvement of algorithms in research.

- **Syntactic tagging**
  Supertaggers have been regarded as components to speed-up full parsers, not as standalone system. However, tagging syntactic functions will be placed more prominently in NLP research, because syntactic functions such as subject and object are of importance for many applications. As I showed in this thesis, syntactic functions can be analyzed accurately using tagging methods.

Automatic assessment

- **Style checking tools**
  So far, style checking tools have often been developed by modeling expert knowledge of editors in the form of style rules. As I showed in this thesis, empirical data can be used as resources to find out how linguistic choice was actually made. This allows for a style checking tool to detect actual practice, i.e. accepted noncompliance of editors with their own guidelines. This will be an interesting area to be studied inter-disciplinarily with corpus linguistics and stylistics.

In summary, this thesis developed an approach to adopt NLP methods to domain-specific texts, and provided a praxis-oriented solution to the automated analysis of syntactic functions in a morphologically rich, free word order language, under the constraint of sparse in-domain data. It can be seen as source and foundation for new interesting questions and approaches around automatic syntactic annotation and assessment.

Appendix A

German Morphosyntactic Tag Set: Gertwol

Gertwol's tag set consists of two types of tags: 1) basic part of speech tags (cf. Table A.1); and 2) morphosyntactic features (cf. Table A.2) (Haapalainen and Majorin, 1994).[1] The basic POS tags are combined with morphosyntactic features. For example, the word *Tischs* is analyzed as `S MASK SG GEN` (cf. (1)). The first tag `S` belongs to the basic POS tag set, and the remaining tags are morphosyntactic features.

(1) - Tischs
"Tisch" S MASK SG GEN

[1]The tag set is based on Haapalainen and Majorin (1994). The tag set is also available at: `http://www2.lingsoft.fi/doc/gertwol/intro/tags.html` (last visited: 2015/08/21)

A1. Basic Part of Speech Tag Set

Type	**Gertwol's Tags**	**Description**
Noun	S, S(A), S(PART)	noun, nominal adjective, nominal participle
Pronoun	PRON	-
Adjective	A, A(PART)	adjective, adjectival participle
Number	NUM	-
Abbreviation	ABK	-
Adverb	ADV	-
Verb	V, PRÄF	verb, separable verbal prefix
Conjunction	iKONJ, nKONJ, sKONJ, uKONJ	infinitive, coordinating, comparative, subordinating conjunction
Preposition	PRÄP	-
Interjection	INTERJ	-

Table A.1: Gertwol's basic POS tag set

A2. Morphosyntactic Features

Type	**Gertwol's Tags**	**Description**
Case	NOM, AKK, GEN, DAT	nominative, accusative, genitive, dative
Gender	MASK, FEM, NEUTR	masculine, feminine, neuter
Number	SG, PL	singular, plural
Number and person	SG1, SG2, SG3; PL1, PL2, PL3	singular 1st person, singular 2nd person, singular 3rd person; plural 1st person, plural 2nd person, plural 3rd person
Style	SELTEN, VERALTET, GESPROCHEN, HÖFLICH, FALSCH	rare, obsolete, colloquial, polite, wrong forms
Features for adjectives	POS, KOMP, SUP, SUP2, STARK, SCHWACH, INV	positive, comparative, superlative, superlative 2, strong declined, weak declined, uninflectable
Features for numbers	KARD, ORD, BRUCH, RÖM	cardinal, ordinal, fraction, Roman numeral
Features for proper nouns	Famname, Vorname	surname, first name
Features for verbs	TRENNBAR, IND, KONJ, IMP, INF, PART, PRÄS, PRÄT, PRÄF, zu, es	separable, indicative, conjunctive, imperative, infinitive, participle, present tense, past tense, perfect, verb form with *zu*, verb form with *es*
Features for prepositions	pre, post; dat, akk, gen	preposition, postposition; dative, accusative, genitive preposition
Features for determiners and/or pronouns	poss, refl, rez, PERS, DEM, RELAT, INTERROG, DEF, INDEF, NEG, DET	possessive, reflexive, reciprocal, personal, demonstrative, relative, interrogative, definite, indefinite, negative, determiner
Features for compounds	ERSTGLIED	first part of compounds

Table A.2: Gertwol's morphosyntactic features

Appendix B

German POS Tag Set: Stuttgart-Tübingen-Tagset (STTS)

POS	Description	Examples
ADJA	attributive adjective	[das] große [Haus]
ADJD	adverbial oder predicative adjective	[er fährt] schnell, [er ist] schnell
ADV	adverb	schon, bald, doch
APPR	preposition, left circumposition	in [der Stadt], ohne [mich]
APPRART	preposition with article	im [Haus], zur [Sache]
APPO	postposition	[ihm] zufolge, [der Sache] wegen
APZR	right circumposition	[von jetzt] an
ART	definite or indefinite article	der, die, das, ein, eine
CARD	cardinal number	zwei [Männer], [im Jahre] 1994
FM	foreign material	[Er hat das mit “] fish [” übersetzt]
ITJ	interjection	mhm, ach, tja
KOUI	subordinating conjunction with “zu” und infinitive	um [zu leben], anstatt [zu fragen]
KOUS	subordinating conjunction with a clause	weil,dass, damit, wenn, ob
KON	coordinating conjunction	und, oder, aber
KOKOM	comparative conjunction	als, wie
NN	common noun	Tisch, Herr, [das] Reisen
NE	proper noun	Hans, Hamburg, HSV
PDS	substituting demonstrative pronoun	dieser, jener
PDAT	attributive demonstrative pronoun	jener [Mensch]
PIS	substituting infinitive pronoun	keiner, viele, man, niemand
PIAT	attributive infinitive pronoun without determiner	kein [Mensch], irgendein [Glas]
PIDAT	attributive infinitive pronoun with determiner	[ein] wenig [Wasser]
PPER	irreflexive personal pronoun	ich, er, ihm, mich, dir
PPOSS	substituting possessive pronoun	meins, deiner
PPOSAT	attributive possessive pronoun	mein [Buch], deine [Mutter]

Table B.1: German POS tag set (STTS) (Schiller et al., 1999)

POS	Description	Examples
PRELS	substituting relative pronoun	[der Hund ,] der
PRELAT	attributive relative pronoun	[der Mann ,] dessen [Hund]
PRF	reflexive personal pronoun	sich, einander, dich, mir
PWS	substituting interrogative pronoun	wer, was
PWAT	attributive interrogative pronoun	welche[Farbe], wessen [Hut]
PWAV	adverbial interrogative or relative pronoun	warum, wo, wann, worüber, wobei
PAV	pronominal adverb	dafür, dabei, deswegen, trotzdem
PTKZU	"zu" before infinitive	zu [gehen]
PTKNEG	negation particle	nicht
PTKVZ	separable particle verb	[er kommt] an, [er fährt] rad
PTKANT	answer particle	ja, nein, danke, bitte
PTKA	particle with adjective or adverb	am [schönsten], zu [schnell]
TRUNC	first part of composition	An- [und Abreise]
VVFIN	finite full verb	[du] gehst, [wir] kommen [an]
VVIMP	imperative full verb	komm [!]
VVINF	infinitive full verb	gehen, ankommen
VVIZU	infinitive full verb with "zu"	anzukommen, loszulassen
VVPP	participle perfect full verb	gegangen, angekommen
VAFIN	finite auxiliar verb	[du] bist, [wir] werden
VAIMP	imperative auxiliar verb	sei [ruhig !]
VAINF	infinitive auxiliar verb	werden, sein
VAPP	participle perfect auxiliar verb	gewesen
VMFIN	finite modal verb	dürfen
VMINF	infinitive modal verb	wollen
VMPP	participle perfect modal verb	gekonnt, [er hat gehen] können
XY	non-words containing special characters	3:7, H2O, D2XW3
$,	comma	,
$.	sentence-final punctuation	. ? ! ; :
$(	sentence-internal punctuation	- [,]()

Table B.2: German POS tag set (STTS) (Schiller et al., 1999)

Appendix C

Constraint Grammar: Technical Details

In Constraint Grammar, rules are defined in terms of if-then conditions. An operation is applied to a targeted tag if the conditions are fulfilled (cf. (1)).

(1) `[OPERATION] [TARGET] IF [CONDITIONS]`

An OPERATION involves the actions of removing implausible morphological analyses or selecting plausible ones in the task of morphosyntactic disambiguation. In the task of syntactic labeling, the assignment of syntactic tags is an operation.

A TARGET is the specification of the targeted cohorts (i.e. morphological readings) for which an operation is carried out. By defining the targets, operations can be executed, restricted to some sets of morphosyntactic features, such as pronouns (`PRON`), relative pronouns (`PRON RELAT`), or nominative singular relative pronouns (`PRON RELAT SG NOM`).

A CONDITION is a constraint imposed by the operation. Conditions are defined in the terms of linear orders of tokens. Current tokens have position 0. The previous tokens are expressed in negative numbers: the previous token has position -1, the token before has position -2, and unlimited forwards tokens have position -1*. Positive numbers denote tokens after the current tokens: the next token has position +1, and the next one has position +2. Unlimited afterwards tokens have position +1*. Unlimited forwards and afterwards tokens are searched endlessly by the current tokens. To stop the search, the stopping criterion BARRIER is defined. Thus, the context window in CG can be freely defined.

With regard to the constraints in CG, the following paragraphs describe the syntax of grammar rules, that is, constraints for morphosyntactic disambiguation and syntactic labeling.

The constraints for morphological disambiguation consist of two operations: `REMOVE` and `SELECT`. `REMOVE` discards impossible analyses directly in formulating the contexts in which they do not occur. In `SELECT` operations, implausible analyses are discarded indirectly in selecting plausible morphosyntactic analyses. `REMOVE` and `SELECT` differ in that the `REMOVE` operator is passive and safe but not effective, whereas the `SELECT` operator is aggressive and risky but effective. Therefore, the `REMOVE` operation is typically applied to wide spectrums, whereas the `SELECT` operator is executed on specific and narrow spectrums.

For example, the word *während* has three morphosyntactic analyses and can be a dative or genitive preposition (`pre PRÄP`) or a subordinating conjunction (`uKONJ`):

(2) *während*
"während" pre PRÄP Gen
"während" pre PRÄP SELTEN Dat
"während" uKONJ

Morphosyntactic disambiguation is defined in contexts where dative and accusative prepositions or subordinating conjunctions do not occur or can occur exclusively. For instance, prepositions occur only if nouns or pronouns are placed afterwards. In constraint grammar, the constraint is formulated as follows:

(3)
```
REMOVE (pre PRÄP)
IF ((NOT 1* (PRO) BARRIER (":")) OR (NOT 1* (S) BARRIER (":")))
```

The constraint states that prepositions (`pre PRÄP`) are removed if they are not followed by any pronouns (`PRO`) or nouns (`S`). The search for `PRO` and `S` is stopped if a colon (":") is encountered. For example, to disambiguate *während* further, subordinating conjunctions (`uKONJ`) are selected if they are followed by finite verbs (`FIN`):

(4)
```
SELECT (uKONJ) IF (1* (FIN))
```

Syntactic tags are assigned to valid cohorts by using the operation `MAP`.[1] The operation `MAP` assigns tags to cohorts if the conditions are matched. The operation `MAP` assigns tags only to cohorts that have not yet been mapped to any syntactic tag. Therefore, the order of constraints matters. For example, in the sentence (5), we want to label a syntactic tag to the token *Täter*. The word *Täter* is a nominative noun that is coordinated with *Täterin*. Therefore, it is assigned to a coordinated element `CJ`. In order that the coordinated noun *Täter* is mapped to `CJ` instead of `SUBJ`, a rule stating that `CJ` is mapped to a nominative noun if it precedes a coordinating conjunction (cf. (6-a)) takes precedent over the rule "`SUBJ` is mapped to a nominative noun (cf. (6-b))."

(5) Erfolgen Herstellung, Erwerb, Einfuhr, Ausfuhr, Durchfuhr oder Besitz ausschliesslich zum Zweck des eigenen Konsums, so bleibt die Täterin oder der *Täter* straflos.

(6) a. `MAP (CJ) TARGET (NOUN) IF (-1* (Coord_Conjunction))`

b. `MAP (SUBJ) TARGET (NOM)`

[1] In the constraint grammar, another type of operation `ADD` is used for labeling. The operation `ADD` assigns a tag whenever conditions are met regardless of whether the current cohorts have already been mapped to any syntactic tag. In this operation, the order of constraints does not matter.

Appendix D

Conversion from Dependency Grammar Relations into Topological Field Relations

In this Appendix, I describe the automatic conversion of topological field labels developed for TüBa-D/Z. Six types of cues are used in the conversion from dependency labels in CoNLL format to topological field labels: lemmata, POS tags, dependency grammar labels, dependency paths (heads and children), and positions (cf. Table D.1). Each topological field label is converted individually using these linguistic annotations on TüBa. Table D.1 provides an overview of the conjoined rules for the conversion. For example, the first column in Table D.1 shows that tokens are analyzed as `LB-V1` if the token is annotated as a finite verb by POS tags and `NEB` as the dependency label *and* if the child is not `LB-VF-CONJ`. With regard to coordination structures, the topological field tags are assigned to the same tags as assigned to the immediately dominated verbal elements if the current tokens are tagged as `KON` or `CJ`.

Topological fields tags	Lemma	POS	Dependency grammar labels	Head	Child	Position
LB-V1	-	FIN	NEB	-	¬LB-VF-CONJ	-
LB-V1	-	VVIMP, VAIMP	-	-	-	-
LB-V1	-	FIN	ROOT, S	-	-	'?' afterwards
RB-V1	-	NONFIN	AUX	LB-V1	-	-
RB-V1	-	ADJD	PRED	LB-V1	-	-
LB-V2	-	FIN	ROOT, S	-	¬LB-VF-CONJ	¬ '?' afterwards
LB-V2	-	FIN	SUBJC, OBJC	-	¬LB-VF-CONJ	-
RB-V2	-	PTKVZ	-	-	-	-
RB-V2	-	NONFIN	AUX	LB-V2	-	after LB-V2
RB-V2	-	ADJD	PRED	LB-V2	-	-
TOP	-	NONFIN	AUX	LB-V2	-	before LB-V2
TOP	-	ADJD	PRED	LB-V2	-	before LB-V2
LB-VF-RC	-	PRELS, PRELAT,	-	-	-	-
LB-VF-RC	-	PWS	-	NEB, REL, SUBJC, OBJC	-	after comma, semicolon and colon
LB-VF-RC	welch, welche, welcher, welches	PWAT	-	NEB, REL, SUBJC, OBJC	-	-
RB-VF-RC	-	FIN	REL	-	-	-
LB-VF-CONJ	-	KOUS	-	-	-	-
RB-VF-CONJ	-	FIN	NEB	-	LB-VF-CONJ	-
RB-VF-CONJ	-	FIN	SUBJC, OBJC	-	LB-VF-CONJ	-
RB-VF-CONJ	-	FIN	S, ROOT	-	LB-VF-CONJ	-
VC	-	NONFIN	AUX	¬LB-V1, ¬LB-V2	-	before verb
VC	-	ADJD	PRED	¬LB-V1, ¬LB-V2	-	before verb
LXCOMP	-	-	KOUI	-	-	-
LXCOMP	comma, semicolon, colon	-	-	-	-	
RXCOMP	-	-	OBJI	-	-	-
RXCOMP	-	VVIZU	NEB, SUBJC, OBJC	-	-	-
RXCOMP	-	VVINF, VAINF, VMINF	NEB, SUBJC, OBJC	-	-	after *zu*

Table D.1: Conversion into topological field tags: FIN = (VVFIN, VAFIN, VMFIN), NONFIN = (VVINF, VVIZU, VVPP, VAINF, VAPP, VMINF, VMPP)

Appendix E

NP Chunking

For the statistical error detection presented in Chapter 6, I developed the prototype of an NP chunker that automatically analyzes two types of noun phrases: base noun phrase (BaseNP) and maximally projected noun phrases (MaxNP). In this appendix, I briefly describe the NP chunker.

A BaseNP is a noun chunk that does not entail more than one noun or pronoun, whereas a MaxNP corresponds to a maximally projected noun phrase.[1] NP chunks are represented in the form of BIO. In BIO chunking, the beginning of a chunk, the intermediate part of a chunk, and the outside of a chunk are marked as B, I, O, respectively. For example, the sentence (1) is analyzed by our NP chunker as (2). A BaseNP chunk corresponds to the tokens assigned to BaseNP-B and BaseNP-I, whereas a MaxNP chunk corresponds to the tokens tagged as MaxNP-B and MaxNP-I.

[1] Relative clauses are excluded from maximally projected noun phrases.

(1) Die mit der europäischen Patentanmeldung erworbenen Rechte bleiben gewahrt.

(2) NP Chunking

Die	O-BaseNP	B-MaxNP
mit	O-BaseNP	I-MaxNP
der	B-BaseNP	I-MaxNP
europäischen	I-BaseNP	I-MaxNP
Patentanmeldung	I-BaseNP	I-MaxNP
erworbenen	B-BaseNP	I-MaxNP
Rechte	I-BaseNP	I-MaxNP
bleiben	O-BaseNP	O-MaxNP
gewahrt	O-BaseNP	O-MaxNP
.	O-BaseNP	O-MaxNP

The NP chunker is processed in the pipeline after supertagging (presented in Chapters 3, 4, and 5) and uses topological dependency grammar labels. BaseNPs are annotated first, then MaxNP.

BaseNP chunking is implemented in a rule-based approach in the framework of Constraint Grammar (CG)(Karlsson et al., 1995). The dependents of nouns, such as determinatives (`DET`) and adjectives (`ATTR`) and nouns are assigned as either B-BaseNP or I-BaseNP, based on topological dependency grammar labels.

In MaxNP chunking, state-of-the-art parsers are used in combination with a rule-based (CG-based) method. MaxNP chunking uses BaseNP annotations. Nouns in BaseNP chunks are categorized into two types, depending on whether their topological dependency labels belong to the arguments of a predicate (cf. (3-a)) or not (i.e. adjuncts, cf. (3-b)). The arguments of a predicate can be the first nouns in the maximally projected noun phrases, whereas the adjuncts (3-b) are the nouns that follow the nouns of the first group within the MaxNP.

(3) a. Arguments such as subjects (`SUBJ` and objects (`OBJA, OBJD, OBJG`) or adverbs such as temporal adverbs (`Zeit`)
b. Adjuncts such as genitive modifiers (`GMOD`), appositions (`APP`), and coordinators (`KON/CJ`)

The dependent of a preposition `PN` is ambiguous. It can belong to the first or second group. Whether the dependent of a preposition `PN` can be the first noun in a MaxNP

depends on the head preposition: it belongs to the first category if the head preposition is attached to a verb, whereas it is categorized into the second category if the head preposition is attached to a noun. Dependency labels of prepositions (`OBJP, PP`) are not directly categorized into noun-attachment or verb-attachment. Instead, they are semantically motivated categorizations between complements (`OBJP`) and adjuncts (`PP`). Complements are attached to verbs, whereas adjuncts can be attached to both verbs and nouns. To distinguish between the two types of attachment in adjuncts (`PP`), a rule-based and a voting-based system are implemented. In the rule-based system, the prepositions that are obviously attached to verbs or nouns are assigned before the voting system is applied: Noun-attachment is more likely if (A) prepositions are between prepositional phrases and (B) prepositions are attached to a noun that is the first noun in the vorfeld. Verb attachment is more likely if prepositions are the first tokens in the vorfeld, the mittelfeld and the nachfeld. Afterwards, two parsers (the Bohnet parser and the ParZu parser used in Chapter 4) are used to disambiguate PP-attachment. The two systems are combined in a simple voting system. If they agree on a PP-attachment type, then it wins. Otherwise, the prediction of the Bohnet parser wins.

Appendix F

Morphosyntactic Ambiguity in German: Qualitative Analysis

In this appendix, I provide a quantitative analysis of the two major problems of ambiguity in German, that is, (i) the coarse ambiguity among functional categories and (ii) the fine-grained morphosyntactic ambiguity in lexical categories. The following sections discuss these two problems in details. Based on this analysis and the one described in Section 4.1, I developed the rule-based supertagger described in Chapter 4.

F.1 Ambiguity of Functional Categories

The following are considered functional categories: determiners, pronouns, adpositions, and conjunctions. The ambiguities among functional categories are classified into three major groups: 1) Determiners vs. pronouns; 2) subordinating conjunctions vs. other morphosyntactic categories; and 3) adpositions vs. other morphosyntactic categories. The following subsections examine each of these major ambiguity groups.

Determiners vs. Pronouns Determiners are ambiguous, and they compete with pronouns (cf. Table F.1). For example, the definite articles *der, die, das* can be demonstrative pronouns or relative pronouns (cf. Example (1)).

Determiners	**Possible morphosyntactic readings (#)**
der, die, das	Determiner, demonstrative pronoun or relative pronoun (3)
dieser, jener	Determiner, demonstrative pronoun (2)
mein, dein, sein, ihr	Determiner or personal pronoun (2)

Table F.1: Morphosyntactic ambiguity: determiners vs. pronouns

(1) a. Determiner *die*:
Die Steuer wird von den Kantonen veranlagt und eingezogen.[1]

b. Demonstrative pronoun *die*:
Die sind alle noch da. (self-created)[2]

c. Relative pronoun *die*:
Personen, **die** Militär- oder Ersatzdienst leisten und dabei gesundheitlichen Schaden erleiden oder ihr Leben verlieren, haben für sich oder ihre Angehörigen Anspruch auf angemessene Unterstützung des Bundes.

Subordinating Conjunctions Most subordinating conjunctions, such as *dass*, *weil*, *obwohl*, *nachdem*, *sobald*, and *solange* are morphologically unambiguous. However, some subordinating conjunctions compete with other morphosyntactic categories, such as prepositions and other types of conjunctions (cf. Table F.2). For example, *während* is ambiguous between a subordinating conjunction and a preposition (cf. Example (2)).

Subordinating conjunctions	**Possible morphological readings (#)**
bis	Subordinating conjunction, preposition, or adverb (3)
damit, so, soweit	Subordinating conjunction or adverb (2)
je, ob, seit, während	Subordinating conjunction or preposition (2)
als	Subordinating conjunction or comparative conjunction (2)
sowie	Subordinating conjunction or coordinating conjunction (2)

TABLE F.2: Morphosyntactic ambiguity of subordinating conjunction

(2) a. Subordinating conjunction *während*:
Forderungen, die entstehen, **während** ein Schiff durch eine andere Person als den Eigentümer betrieben wird, geniessen gleichwohl ein gesetzliches Pfandrecht, es sei denn, das Schiff sei dem Eigentümer durch eine unerlaubte Handlung entzogen worden und der Gläubiger sei nicht in gutem Glauben gewesen.

b. Preposition *während*:
Das unmündige Kind erwirbt das Kantons- und Gemeindebürgerrecht des Vaters, wenn dieser die Mutter heiratet oder **während** der Ehe Schweizer Bürger wird.

[1] All examples in this appendix are derived from Swiss German-language law texts (cf. Section 1.5) unless they are marked otherwise.
[2] This example is not derived from Swiss German-language law texts because it is typically found in colloquial speech.

Adpositions The majority of adpositions are ambiguous. Only a few adpositions are unambiguous, such as *aufgrund*. Adpositions compete with verbal prefixes, conjunctions, and adverbs (cf. Table F.3). For example, *auf* is ambiguous among prepositions, verbal prefixes, and adverbs (cf. Example (3)).

Adpositions	**Possible morphological readings (#)**
um	Preposition, infinite conjunction, verbal prefix, or adverb (4)
entlang, nach	Preposition, verbal prefix, adverb, or postposition (4)
ab, an, auf, durch, mit, über, zu	Preposition, verbal prefix, or adverb (3)
ausser	Preposition, infinite conjunction, or coordinating conjunction (3)
statt [3]	Preposition, infinite conjunction, or verbal prefix (3)
ohne	Preposition, infinite conjunction
für	Preposition or adverb (2)
wegen	Preposition or postposition (2)
bei, gegen, hinter, neben, unter, wider, zwischen	Preposition or verbal prefix (2)

TABLE F.3: Morphosyntactic ambiguity of prepositions

(3) a. Preposition *auf*:
Dieses Gesetz findet keine Anwendung **auf**: das Verfahren von Behörden im Sinne von Artikel 1 Absatz 2 Buchstabe e, soweit gegen ihre Verfügungen die Beschwerde unmittelbar an eine Bundesbehörde unzulässig ist; [...]

b. Verbal prefix *auf*:
Sie ordnet den Aktivdienst an und bietet dafür die Armee oder Teile davon **auf**.

c. Adverb *auf*:
Wie lange werden die Läden heute **auf** sein (Dudenredaktion, 1999, pp. 315)[4]

F.2 Ambiguity of Lexical Categories

The following are considered lexical categories: verbs, nouns, adjectives, and adverbs. As illustrated in Table 4.4, they are ambiguous between each other. In particular, verbs

[4]This example is not derived from Swiss German-language law texts, as the use is typically in colloquial speech. In the Duden, the word *auf* in the example is analyzed as adverb.

with an inseparable prefix such as *ver-*, *be-* or *ent-* are competing with attributive adjectives (e.g. *'bestimmte' Menge*) or adverbs (e.g. *'bestimmt' gute Sache*) (cf. Table F.4). In addition, nouns are competing with other lexical categories if they are placed at the beginning of sentences. At this position, all words are capitalized and the unique capitalization feature of nouns gets lost. For example, the word *Befinden* is ambiguous between the verbs and nouns in the sentence (4).

Verbs	**Examples**	**Possible morphological readings (#)**
PartPast-root	*bestimmt, bedingt, geschützt, gestützt etc.*	Verb or adverb (2)
PartPast-root + *-e* (inseparable prefix)	*verkaufte, beauftragte etc.*	Verb or adjective (2)
PartPast-root + *-en* (inseparable prefix)	*verkauften, beauftragten etc.*	Verb or adjective (2)
v-root + *-en*	*gleichen etc.*	Verb or adjective (2)
v-root + *-en* (Capitalized)	*Beauftragen, Verfahren etc.*	Verb or noun (2)

TABLE F.4: Morphosyntactic ambiguity of verbs

(4) **Befinden** sie sich im Ausland, so erteilt der Kanton, in dem der Zirkus oder die fahrende Tierschau erstmals gastieren will, die Bewilligung, soweit nötig unter Berücksichtigung der Einfuhrbewilligung des BVET.

Moreover, verbs, nouns, and adjectives are ambiguous at the following fine-grained morphosyntactic levels:

- Verbs: person, number, aspect, tense, and mood
- Nouns: case, gender, and number
- Adjectives: case, gender, and number

Verbs Syncretism is common in German verb paradigms. In particular, the inflectional exponents *-en* and *-t* are ambiguous in finiteness. The inflectional ending *-en* marks indicative or conjunctive first person plural (e.g. *wir trink-en*), indicative or conjunctive third person plural (e.g. *sie trink-en*), or infinitive (e.g. *zu trink-en*). In addition, inseparable prefix verbs, such as verbs with e.g. *be-, ent-* can be past participles (e.g. 'bekommen', 'entkommen'). The inflectional ending *-t* can be either indicative or imperative in the second person plural (e.g. *ihr trink-t* or *Trinkt!*). It can be indicative in

the third person singular (e.g. *er trink-t*). Inseparable verbs can be also past participles (e.g. *verkauft* or *bekämpft*).

Nouns As stated by Eisenberg et al. (1995, pp. 219f), German nouns are classified into 10 declination types according to gender (i.e. masculine, feminine, and neuter), number (i.e. singular and plural) and case (i.e. nominative, accusative, dative, and genitive). Some nouns are associated with more than one gender class according to the different meanings (e.g. *der Leiter* vs. *die Leiter*) or regional preferences (e.g. *der Tunnel* in Germany vs. *das Tunnel* in Switzerland). Depending on the number class (i.e. ∅, *-e, -er, -(e)n, -(e)s*), nouns either do not inflect (e.g. *Onkel, Onkel*) or inflect (e.g. *Jahr, Jahre; Kind, Kinder; Frau, Frauen; Handy, Handys*). With regard to case, the majority of German nouns do not exhibit an overt morphological marking. Plural nouns are marked with *-n* in the dative case, whereas singular nouns are inflected with *-(e)s* in the masculine and neuter genders.[5] Otherwise, case is morphologically unmarked. The dependents of nouns, such as determiners and adjectives, undergo inflection more distinctively. However, morphologically, they are not fully specified.

Adjectives In German, attributive adjectives belong to a strong, weak, or mixed declension class. Adjectives are declined strongly if determiners are absent. They are declined weakly or mixed, depending on the declension class of determiners. In the strongly declension class, the adjectives are declined analogically to determiners. In the weakly and mixed declension class, the adjectives have less distinctive forms for case, gender, and number.

[5] In a weakly inflecting masculine class (e.g. *Junge, Student*), the nouns inflect with *-(e)n* if they are singular and accusative or dative and genitive; they inflect with *-(e)n* if they are plural.

Appendix G

ML-Based Tagging Methods: Technical Details

This appendix outlines the technical background of the machine learning methods used for tagging problems, particularly the Hidden Markov Model (HMM), the Maximum Entropy Markov Model (MEMM), the Conditional Random Fields (CRF) model, and the structured perceptron. The HMM was used as the basic statistical tagger in the evaluation of the rule-based supertagger in Chapter 4. I use the CRF and the structured perceptron in Chapter 5. Here, I outline the MEMM for the better understanding of the development of the CRF algorithm. In this appendix, these four tagging methods are described with regard to their algorithms, decoding, parameter estimation, and learning. If readers are interested in how a statistical tagging method works but not in decoding and learning algorithms, I suggest reading the paragraph "model" of each method and skip the parts about decoding, parameter estimation, and learning. If readers are not interested in the technical background of statistical tagging methods but are interested in their application, I suggest skipping this appendix and moving directly to the description of their implementation provided in Section 5.3.

G.1 Hidden Markov Model (HMM)

The Hidden Markov Model (HMM) is a finite state automaton with stochastic state transitions and observation emissions (Rabiner, 1989) (cf. Figure G.1). The following review is based on Smith (2011, chap. 3.3.3), Jurafsky and Martin (2009, chap. 6.1-6.5), Koller and Friedman (2009, chap. 6.2), Manning and Schütze (1999, chap. 9).

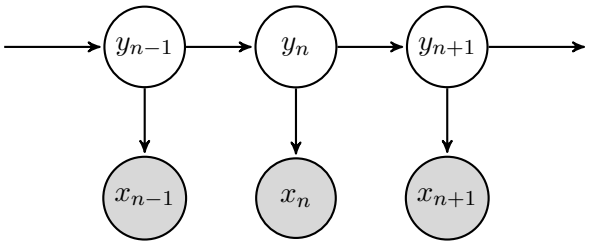


FIGURE G.1: HMM: x = observation, y = state, n = time: x is observable, which is marked with gray

Model Formally, the HMM models the joint probability of an input sequence, $x_1 \dots x_n$, i.e. $x_{1:N}$ and an output sequence $y_1 \dots y_n$ i.e. $y_{1:N}$. The goal of HMM is to find the optimal output sequence $y_{1:N}$, given $x_{1:N}$, such that the joint distribution $P(x_{1:N}, y_{1:N}.)$ is maximized:

$$f(x_{1:N}) = \arg\max_{y_{1:N}} P(x_{1:N}, y_{1:N}) \tag{G.1}$$

By using the chain rule, we can decompose the joint probability by using Equation (G.2).[1]

$$\begin{aligned} P(x_{1:N}, y_{1:N}) &= P(y_{1:N})P(x_{1:N}|y_{1:N}) \\ &= P(y_1 \dots y_N)P(x_1 \dots x_N|y_1 \dots y_N) \\ &= P(y_1)P(y_2|y_1) \dots P(y_N|y_1, y_2, y_3 \dots y_{N-1}) \times P(x_1 \dots x_N|y_1 \dots y_N) \end{aligned} \tag{G.2}$$

To make the computation of joint probability $P(x_{1:N}, y_{1:N})$ tractable, HMM makes two assumptions: first, each observation x_n is only dependent on the current state y_n. Second, each state y_n is dependent only on some previous states, such as y_{n-1}. This independence assumption is called the Markov assumption. Under these assumptions, the HMM is used to reformulate the joint probability as Equation (G.3).

$$P(x_{1:N}, y_{1:N}) = \prod_{n=1}^{N} P(y_n|y_{n-1})P(x_n|y_n) \tag{G.3}$$

[1]The chain rule of conditional probability is applied as follows:

$$P(\alpha_1 \dots \alpha_k) = P(\alpha_1)P(\alpha_2|\alpha_1) \dots P(\alpha_k|\alpha_1 \cap \dots \cap \alpha_{k-1})$$

The HMM (G.3) is a first order Markov model in which the state y_n is dependent only on the previous state y_{n-1}. By extending the conditional term, we can define a nth order model. In a second order Markov model or trigram Markov model, for example, the conditional term is extended to two previous states y_{n-1} and y_{n-2}:

$$P(x_{1:N}, y_{1:N}) = \prod_{n=1}^{N} P(x_n|y_n)P(y_n|y_{n-1}, y_{n-2}) \tag{G.4}$$

The HMM is sometimes called state-observation model (Rabiner, 1989).[2] As the formula (G.3) can be decomposed into Equation (G.5), HMM can be interpreted as two separate operations: observation probability $P(x_n|y_n)$; and state transition probability $P(y_n|y_{n-1})$. Observation probability (or emission probability) $P(x_n|y_n)$ is the probability of the emission of current state y_n on current observation x_n. Transition probability $P(y_n|y_{n-1})$ represents the transition of two successive states.

$$P(x_{1:N}, y_{1:N}) = \prod_{n=1}^{N} P(x_n|y_n) \times \prod_{n=1}^{N} P(y_n|y_{n-1}) \tag{G.5}$$

In NLP, the start position and the end position of a sentence often play an important role. The start and end states can be added to the model as follows:

2

From the point of view of graphical models, HMM is a dynamic Bayesian network model (DBN) that describes the joint probability distribution (Koller and Friedman, 2009):

$$P(v_{1:n}) = \prod_{n=1}^{N} P(v_n|Pa_{v_n})$$

The function Pa maps the variable v_n to the parent variable, that is, the source of incoming edges to v_n in a graphical model (Figure G.1). By assuming that the child node is only dependent on its parent node, HMM represents the joint probability as the composition of the child-mother conditional probability, $P(x_n|y_n)$ and $P(y_n|y_{n-1})$, in a sequential model:

$$\begin{aligned} P(x_{1:N}, y_{1:N}) &= \prod_{n=1}^{N} P(x_n|Pa_{x_n} = y_n)P(y_n|Pa_{y_n} = y_{n-1}) \\ &= \prod_{n=1}^{N} P(x_n|y_n)P(y_n|y_{n-1}) \end{aligned}$$

The formula states that each state y_n at time t_n moves to the next state y_{n+1} at time t_{n+1} until the end state y_{N+1}. The current state y_n at time t_n has a direct effect on the next state y_n. Because the model can be seen as the trajectory of the time splice, t_{n-1} to t_n, HMM is sometimes called the temporal Bayesian network model.

$$P(x_{1:N}, y_{1:N}) = P(y_1|START) \times \prod_{n=1}^{N} P(x_n|y_n)P(y_{n+1}|y_n) \tag{G.6}$$

If $x_{1:N}$ is equal to a sentence *Meine Katze miaut* and $y_{1:N}$ is equal to the tag sequence `DET, SUBJ, LB-V2`, then the joint probability $P(x_{1:N}, y_{1:N})$ is estimated by applying the equation (G.6):

$$P(x_{1:N}, y_{1:N}) = P(DET|START) \times P(Meine|DET) \times P(SUBJ|DET) \times P(Katze|SUBJ) \times P(LB-V2|SUBJ) \times P(miaut|LB-V2) \times P(END|VB-V2) \tag{G.7}$$

Decoding Since HMM is a sequential model, we need to not only find the best y_n for x_n, but also the best sequence $y_1 ... y_n$ for $x_1 ... x_n$:

$$\begin{aligned} f(x_{1:N}) &= \underset{y_1 \ldots y_{n+1} \in Y}{\arg\max} P(x_{1:N}, y_{1:N}) \\ &= \underset{y_1 \ldots y_{n+1} \in Y}{\arg\max} \prod_{n=1}^{N} P(x_n|y_n)P(y_n|y_{n-1}) \end{aligned} \tag{G.8}$$

The decoding problem is to find the optimal $y_{1:N}$, that is, $\arg\max_{y_1 \ldots y_{n+1} \in Y}$. To find the most likely hidden sequence $y_{1:N}$, we use a search algorithm, such as the Viterbi algorithm, beam search, and A* search. In this thesis, I focus on the Viterbi algorithm. Viterbi decoding finds the best two sequential states backwards and recursively. To find the best previous state, we keep two registers: path score and back-pointer. The path score is stored in a dynamic programming table in which each cell with time n and current state k is recursively filled with the current state y_n^k (kth tag candidate of nth token) and its previous state y_{n-1}^j (jth tag candidate of *n-1*th token). The path score is computed as shown in Equation (G.9). The three factors multiplied in Equation (G.9) are (A) the previous path score: $\pi(n-1, y_{n-2}^i, y_{n-1}^j)$, (B) the transition probability from the previous state y_{n-1}^j to y_n^k: $P(y_n^k|y_{n-1}^j)$, and (C) the emission probability of the input x_n given the current state y_n^k: $P(x_n|y_n^k)$.

$$\pi(n, y_{n-1}^j, y_n^k) = \max_{0 \le j \le N} \pi(n-1, y_{n-2}^i, y_{n-1}^j) \times P(y_n^k|y_{n-1}^j) \times P(x_n|y_n^k) \tag{G.9}$$

Because the transition from *n-1* to n has as many possible states as the number of states of time *n-1*, we need to maximize the value of the path score for the current state, y_n^k.

The initial path score, $\pi(0, y^j_{n-1}, y^k_n)$, is set at 1, and the path score is computed from the first to nth token. At each token, we store the most likely previous state y^j_{n-1} for each current state y^k_n. The most likely previous state y^j_{n-1} is the one that maximizes the path score for the current state y^k_n. For this purpose, we have an extra register, which is called *back-pointer* (BP) (cf. Equation (G.10)). By registering the best path backwards, the most probable previous tag y^j_{n-1} is recursively pointed back from the current state y^k_n. When we reach the end state of a sequence and know which previous state at time N leads to the end state with the highest path score, we trace the most probable path backwards by using the back-pointer. The backtracking is recursively carried out from the end state to the initial state.

$$BP(n, y^j_{n-1}, y^k_n) = \underset{1 \leq j \leq J}{\arg\max}\, \pi(n-1, y^i_{n-2}, y^j_{n-1}) \times P(y^k_n | y^j_{n-1}) \times P(x_n | y^k_n) \tag{G.10}$$

Parameter Estimation and Learning For the estimation of the emission probability $P(x_n|y_n)$ and transition probability $P(y_n|y_{n-1})$, we use maximum likelihood estimation (MLE). By using an annotated corpus, we count the relative frequency of the co-occurrence of observation-state pairs for $P(x_n|y_n)$ and state sequence pairs for $P(y_n|y_{n-1})$:

$$P(y_n|y_{n-1}) = \frac{C(y_n, y_{n-1})}{C(y_{n-1})} \tag{G.11}$$

$$P(x_n|y_n) = \frac{C(x_n, y_n)}{C(y_n)} \tag{G.12}$$

If we have x_n equal to *Katze*, y_n equal to `SUBJ` and y_{n-1} equal to `DET`, then MLE is calculated by applying Equation (G.11) and (G.12) to them as Equation (G.13) and (G.14), respectively. Equation (G.13) and (G.14) are computed by counting the occurrences of the tag sequence `DET-SUBJ`, the tag `DET`, the word-tag pair (*Katze*, `SUBJ`), and the tag `SUBJ` in the annotated corpus:[3]

[3] Because a training corpus is limited and cannot contain all vocabularies, and the parameter estimation is operated by multiplication, there are some issues of null occurrence and zero probability. This sparse data problem or unknown word issue is handled by using smoothing techniques, such as Laplace, Good-Turning estimation, interpolation, and back-off (Chen and Goodman, 1998). For instance, Laplace smoothing adds one to the count of the numerator and vocabulary V (the set of all words) to the denominator.

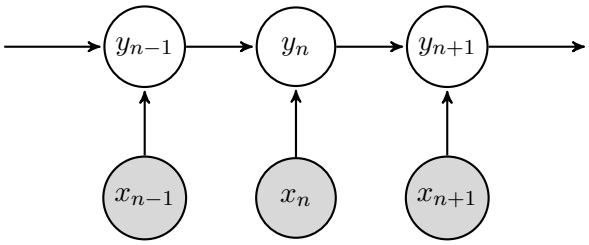


FIGURE G.2: MEMM: x = input, y = output: the input is observable, which is marked with gray

$$P(SUBJ|DET) = \frac{C(SUBJ, DET)}{C(DET)} \tag{G.13}$$

$$P(Katze|SUBJ) = \frac{C(Katze, SUBJ)}{C(SUBJ)} \tag{G.14}$$

G.2 Maximum Entropy Markov Model (MEMM)

The Maximum Entropy Markov Model (MEMM) is a variant of maximum entropy models (MaxEnt), that is, log linear models, which are especially geared to sequential models. The key characteristic of MEMM is the use of features (cf. Figure G.2). The following review is based on Smith (2011, pp. 204ff), Jurafsky and Martin (2009, chap. 6.8), McCallum et al. (2000).

Model The Maximum Entropy Markov Model (MEMM) proposed by McCallum et al. (2000) is a discriminative model that learns the posterior probability $P(y_{1:n}|x_{1:n})$ directly. The posterior probability is decomposed as Equation (G.15). $P(y_n|x_n, y_{n-1})$ provides the probability of the current state y_n, given the previous state y_{n-1} and the current observation x_n. The previous state and the observation are treated as a condition or evidence in the current state (McCallum et al., 2000). MEMM can be seen as modeling the transition probability $P(y_n|x_n, y_{n-1})$, given y_{n-1} on x_n.

$$P(y_{1:N}|x_{1:N}) = \prod_{i=1}^{n} P(y_n|x_n, y_{n-1}) \tag{G.15}$$

To encode the transition probability $P(y_n|y_{n-1}, x_n)$, MEMM uses MaxEnt (cf. Equation (G.17)).[4] The weight (or parameter) w_d is learned during training. $f_d(x_n, y_n)$ is an dth feature function of input x_n and output y_n. The feature function is a binary indicator function that signals the existence and non-existence of a feature.

$$P(y_n|y_{n-1}, x_n) = \frac{1}{Z(x, y_{n-1})} \exp\left(\sum_{d=1}^{D} w_d f_d(x_n, y_n)\right) \tag{G.17}$$

For instance, the 100th feature function (G.18), $f_{100}(x_n, y_n)$, is defined such that value 1 is assigned if the current word is *Katze* and the output is `SUBJ`(ect). Features capture a range of cues for the identification of a class in the form of a template '*the observation of* x_n *and the target class* y_n'.

$$f_{100}(x_n, y_n) = \begin{cases} 1 & \text{if } x_n \text{ is } \textit{Katze} \text{ and } y_n \text{ is } \texttt{SUBJ} \\ 0 & \text{otherwise} \end{cases} \tag{G.18}$$

The parameters $w_0 \dots w_d$ are linearly combined with feature vectors, as shown in Equation (G.19).

$$W * F = \sum_{d=0}^{D} w_d f_d(x_n, y_n) = w_0 + w_1 \times f_1 + w_2 \times f_2 + \dots + w_d \times f_d \tag{G.19}$$

The dot product $W * F$ is taken into exponential (cf. Equation (G.20)).[5] We now have a value greater than 0. The exponential dot product $exp(W * F)$ is between 0 and 1 if the dot product has a negative value. The value is above 1 if the dot product has a positive value.

$$\exp(W * F) = \exp\left(\sum_{d=0}^{D} w_d f_d(x_n, y_n)\right) \tag{G.20}$$

[4] The maximum entropy model has the form of a log-linear or exponential function:

$$P(y|x) = \frac{1}{Z} \exp\left(\sum_{d=0}^{D} w_d f_d(x, y)\right) \tag{G.16}$$

[5] As usual, $exp(x) = e^x$.

To ensure that the exponential dot product is a well-formed probability distribution, we ensure that the sum of the probability $P(y_n|y_{n-1}, x_n)$ is 1 (cf. Equation (G.21)). For this purpose, the normalization term Z is added in Equation (G.17). Z is the denominator of the probability estimation of $P(y_n|y_{n-1}, x_n)$ and the sum over all the values of the exponential dot product of the possible outputs Y_n for given y_{n-1} and x_n (cf. Equation (G.22)). By applying Equation (G.22), Equation (G.17) can be reformulated as Equation (G.23).

$$\sum_{y_n \in Y_n} P(y_n|x_n, y_{n-1}) = 1 \tag{G.21}$$

$$Z(x, y_{n-1}) = \sum_{y_n \in Y_n} \exp\left(\sum_{d=0}^{D} w_d f_d(x_n, y_n)\right) \tag{G.22}$$

$$P(y_n|y_{n-1}, x_n) = \frac{1}{Z(x, y_{n-1})} \exp\left(\sum_{d=1}^{D} w_d f_d(x_n, y_n)\right) = \frac{\exp\left(\sum_{d=0}^{D} w_d f_d(x, y)\right)}{\sum_{y \in Y} \exp\left(\sum_{d=0}^{D} w_d f_d(x_n, y_n)\right)} \tag{G.23}$$

For example, we have the sequence of tokens, *Die, Zürcher, Verlobung.* We have the current input token *Zürcher* and the previous state `DET`. For example, there are two possible dependency grammar tags `Y = {ATTR, SUBJ}` for the current state. To calculate Z, the exponential dot product of `ATTR` and `SUBJ` are summed. This sum is used as the denominator for the estimation of the conditional probability $P(y_n|y_{n-1}, x_n)$. The nominator is the exponential dot product for each class y_n, i.e. `ATTR` and `SUBJ`.

Decoding To find the best sequence $y_1 \dots y_m$, we can use Viterbi-like decoding. Because we defined the model $P(y_n|x_n, y_{n-1})$, the decoding is formulated as follows:

$$\pi(n, y_{n-1}^j, y_n^k) = \max_{0 \le j \le J} \pi(n-1, y_{n-2}^i, y_{n-1}^j) \times P(y_n^k|x_n, y_{n-1}^j) \tag{G.24}$$

The path score is the maximal value of the multiplication of (i) the path score of the previous output y_{n-1} with (ii) the current score. The best scoring of the previous output y_{n-1}^j among J possible outputs at time t_{n-1} is backtracked by the back-pointer (BP):

$$BP(n, y_{n-1}^j, y_n^k) = \arg\max_{1 \le j \le J} \pi(n-1, y_{n-2}^i, y_{n-1}^j) \times P(y_n^k|x_n, y_{n-1}^j) \tag{G.25}$$

Parameter Estimation and Learning In MEMM, we learn the weights $w_1 \dots w_m$ of the feature $f_1 \dots f_m$ from the training set, $(x_1, y_1) \dots (x_i, y_i)$. To estimate the weights, we use conditional maximum likelihood estimation. By reformulating the probability $P(y_n|y_{n-1}, x_n)$ into $P_{y_{n-1}}(y_n|x_n)$, MEMM can be trained similar to MaxEnt.

For the entire training set $(x_{1:I}, y_{1:I})$, we want to find the log likelihood of the conditional probability, given weight vector w' (cf. Equation (G.26)). Furthermore, we want to find the value of w' that maximizes the function $l(w)$ and best fits the entire training set (cf. Equation (G.27)).

$$l(w') = \sum_i \log P_{y_{n-1}^i}(y_n^i|x_n^i) \tag{G.26}$$

$$w' = \arg\max_{w'} l(w') = \arg\max_{w} \sum_i \log P_{y_{n-1}^i}(y_n^i|x_n^i) \tag{G.27}$$

However, the ultimate goal of the learning is a generalization. To handle unseen data generally, we add a regularization term as the quadratic function of parameters $\lambda \sum_{m=1}^{M} w_m^2$. The regularization avoids overfitting, that is, fitting the training data too well (cf. Equation (G.28)). By subtracting the regularization term and creating a new function $L(w)$, $l(w)$ remains smaller and the parameter remains not too high. λ is the regularization parameter that controls the magnitude of overfitting.

$$w = \arg\max_{w} L(w) = \arg\max_{w} \sum_i \log P_{y_{n-1}^i}(y_n^i|x_n^i) - \lambda \sum_{m=1}^{M} w_m^2 \tag{G.28}$$

To find the optimal weight in MaxEnt, we use an iterative scaling and gradient-based learning method (Malouf, 2002). For example, in MEMM, McCallum et al. (2000) used Generalized Iterative Scaling (GIS) (Darroch and Ratcliff, 1972). In GIS, each parameter is iteratively updated based on the ratio of the expected value of parameters and the empirical value of parameters until $L(w)$ reached convergence.

G.3 Conditional Random Fields (CRF)

The Conditional Random Field (CRF) is another conditional probabilistic model used for sequences but with global normalization (cf. Figure G.3). This section is based on

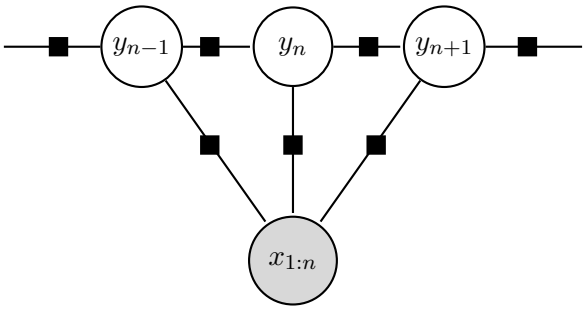


FIGURE G.3: CRF: $x_{1:n}$ = input, y = output; x is observable, which is marked with gray. The rectangle in the edge indicates the factorization

Sutton and McCallum (2011), Smith (2011), Koller and Friedman (2009), Lafferty et al. (2001).[6]

Model CRF is a random field model (also known as undirected graph or Markov network) used for conditional probability $P(y_{1:N}|x_{1:N})$. A Markov network consists of nodes and undirected edges. An edge binds two nodes and represents a direct probabilistic interaction between the nodes, that is, variables. Because the edges are not directed, we need to capture locally the interactions between the variables. Therefore, we localize (i.e. factorize) the variables and define a factor $\phi(D)$ (Koller and Friedman, 2009, pp. 143). The factor is a function, of which the scope D is the set of local variables that interact directly. As in a Bayesian network, the local interactions of variables are combined by multiplication for the global model. Thus, a Markov network encodes the following conditional probability distribution (cf. Equation (G.29)). The normalization term Z sums all possible values of the sequence $y_{1:n}$ (cf. Equation (G.30)).

$$P(y_{1:n}|x_{1:n}) = \frac{1}{Z(x_{1:n})} P'(x_{1:n}, y_{1:n}) = \frac{1}{Z(x_{1:n})} \prod_{m=1}^{M} \phi_m(D_m) \tag{G.29}$$

$$Z(x_{1:n}) = \sum_{y_{1:n}} P'(x_{1:n}, y_{1:n}) \tag{G.30}$$

Because we have a sequence model, we assume that the sequence of $y_{1:n}$ is a line and is connected. This type of CRF is sometimes called a linear-chain CRF (Sutton and

[6] In addition, I used the course materials of M. Collins http://www.cs.columbia.edu/~mcollins/ (last visited: 2015/06/30)

McCallum, 2011). In a linear-chain first-order CRF, for example, the direct interacting variables are the current output y_n, the previous output y_{n-1}, the sequence $x_{1:n}$ and the time n in a Markov network (cf. Equation (G.31)). Note that the number of factor ϕ corresponds to the number of time slices, that is, the number of tokens in sentences, N.

$$P(y_{1:n}|x_{1:n}) = \frac{1}{Z(x_{1:n})} \prod_{n=1}^{N} \phi_n(x_{1:n}, y_{n-1}, y_n, n) \tag{G.31}$$

The factor function ϕ_n is the exponential of the sum of weighted feature functions (cf. Equation G.32). The local feature function f_i maps features into feature vectors. The scopes of features are the current output y_n, the previous output y_{n-1}, the entire input sequence $x_{1:n}$, and the time n. The number of features is I. As we have already seen in MEMM, we can define features as observation-target combinations.

$$\phi_n(x_{1:n}, y_{n-1}, y_n, n) = \exp(\sum_{d=1}^{D} w_d f_d(x_{1:n}, y_{n-1}, y_n, n)) \tag{G.32}$$

We now move the exponential outside the potential in Formula (G.31).

$$\begin{aligned} P(y_{1:n}|x_{1:n}) &= \frac{1}{Z(x_{1:n})} \prod_{n=1}^{N} \exp(\sum_{d=1}^{D} w_d f_d(x_{1:n}, y_{n-1}, y_n, n)) \\ &= \frac{1}{Z(x_{1:n})} \exp\left(\sum_{n=1}^{N}\sum_{d=1}^{D} w_d f_d(x_{1:n}, y_n, y_{n-1}, n)\right) \\ \textit{where} \quad Z(x_{1:n}) &= \sum_{y_{1:n}} \exp\left(\sum_{n=1}^{N}\sum_{d=1}^{D} w_d f_d(x_{1:n}, y_n, y_{n-1}, n)\right) \end{aligned} \tag{G.33}$$

Thus, in CRF, we build a MaxEnt model for the entire time $t_{n:1}$ and normalize the conditional probability distribution $P(y_{1:n}|x_{1:n})$ for the whole sequence. Hence, CRF solved a *label bias problem* (Lafferty et al., 2001). The label bias problem exists because the current state followed by a state with a few outgoing edges gets a higher score than it should. The problem is that the decision about the state y_n is made by taking into account the local per-state probability distribution in MEMM. In MEMM, we build a MaxEnt model at each time t_n, and the conditional probability distribution is normalized by the partition function Z. MEMM calculates the transition probability from state y_{n-1}^j to y_n^k, which is normalized locally only against all possible K states at time n, $y_n^1 \dots y_n^K$,

given y_{n-1}^{j}. Thus, the state y_n^k achieves the maximal score, that is, 1, if it is the single ongoing state from the previous state. Moreover, in decoding, we have seen that the score of the previous states is calculated such that the current score becomes much higher if the previous state score is high. This causes a biased prediction in MEMM. CRF is an improved algorithm to use in solving this label bias problem. CRF solves the problem by global normalization against all transitions. Therefore, the transition from state to state is not biased by the number of local edges.

Decoding The goal of a linear-chain CRF is to find the best sequence $y_1 \dots y_n$:

$$\arg\max_{y_{1:n} \in Y_{1:n}} P(y_{1:n}|x_{1:n}) \tag{G.34}$$

In a linear-chain CRF, we compute the probability at time $t_{1:n}$ as (G.35). The local feature function can be defined as ϕ'_n (cf. Equation (G.36)). The Equation (G.35) can be thus formulated as (G.37).

$$P(y_{1:n}|x_{1:n}) = \frac{1}{Z(x_{1:n})} \exp\left(\sum_{n=1}^{N}\sum_{d=1}^{D} w_d f_d(x_{1:n}, y_n, y_{n-1}, n)\right) \tag{G.35}$$

$$\phi'_n(x_{1:n}, y_n, y_{n-1}, n) = \sum_{d=1}^{D} w_d f_d(x_{1:n}, y_n, y_{n-1}, n) \tag{G.36}$$

$$P(y_{1:n}|x_{1:n}) = \frac{1}{Z(x_{1:n})} \exp\left(\sum_{n=1}^{N} \phi'_n(x_{1:n}, y_{n-1}, y_n, n)\right) \tag{G.37}$$

By removing the denominator and the exponential, we compute the maximal value of a sequence (cf. Equation (G.38)). To find the arg max by the use of Viterbi, we want to know the maximal value of the path score at time t_n. To segment the computation into time slice t_n, we remove the sum over all sequences as (G.39). Thus, we compute the dot product at each time t_n to derive the maximal scoring y_n^j.

$$\arg\max_{y_{1:n} \in Y_{1:n}} \sum_{n=1}^{N} \phi'_n(x_{1:n}, y_{n-1}, y_n, n) \tag{G.38}$$

$$\arg\max_{y_{1:n} \in Y_{1:n}} \phi'_n(x_{1:n}, y_{n-1}, y_n, n) \tag{G.39}$$

Viterbi-like decoding is formulated (cf. Equation (G.40)). The path score is the maximal value of the sum of (i) the path score of the previous output y_{n-1} and (ii) the current score. The best scoring previous output y^j_{n-1} among N possible outputs at time t_{n-1} is backtracked by the back pointer (BP) (cf. Equation (G.41)). By using Viterbi, we can implement the decoding task simply and efficiently.

$$\pi(n, y^j_{n-1}, y^k_n) = \max_{0 \le j \le N} \pi(n-1, y^i_{n-2}, y^j_{n-1}) + \phi'_n(x_{1:n}, y_{n-1}, y_n, n) \tag{G.40}$$

$$BP(n, y^j_{n-1}, y^k_n) = \arg\max_{1 \le j \le N} \pi(n-1, y^i_{n-2}, y^j_{n-1}) + \phi'_n(x_{1:n}, y_{n-1}, y_n, n) \tag{G.41}$$

Parameter Estimation and Learning In CRF, we learn the weights (or parameters) $w_1 \dots w_m$ of features $f_1 \dots f_m$ from a training set, $(x^1_{1:n}, y^1_{1:n}) \dots (x^T_{1:n}, y^T_{1:n})$. To estimate the parameters, we use the conditional maximum likelihood estimation just like MEMM. For the training set $(x^1_{1:n}, y^1_{1:n}) \dots (x^T_{1:n}, y^T_{1:n})$, we want to maximize the conditional log likelihood $l(w)$ (cf. Equation (G.42)). The maximization of this log-likelihood is performed to find the parameters that best fit the training data (cf. Equation (G.43)).

$$l(w) = \sum_{t=1}^{T} \log P(y^t_{1:n} | x^t_{1:n}) \tag{G.42}$$

$$w = \arg\max_w l(w) \tag{G.43}$$

To avoid over-fitting, the conditional log likelihood takes the regularization (cf. Equation (G.45)). The regularization term *lambda* is $1/2\sigma^2$, which determines the magnitude of the penalty for over-fitting. Thus, the goal of the learning is to find the optimal weight w that maximizes the regularized conditional log-likelihood (cf. Equation (G.45))

$$L(w) = \sum_{t=1}^{T} \log P(y^t_{1:n} | x^t_{1:n}) - \lambda \sum_{m=1}^{M} w^2_m \tag{G.44}$$

$$w' = \arg\max_w L(w) = \arg\max_w \sum_{t=1}^{T} \log P(y^t_{1:n} | x^t_{1:n}) - \lambda \sum_{m=1}^{M} w^2_m \tag{G.45}$$

G.4 Structured Perceptron

The structured perceptron algorithm proposed by Collins (2002) is a feature-rich, non-probabilistic, linear hyperplane learning algorithm used for sequences. A hyperplane classifier divides a geometric space into N classes. This subsection is based on Hagan et al. (2015), Daumé III (2014), Collins (2002).[7]

Model: Standard Perceptron - Binary Classification The perceptron learning algorithm was originally proposed by the psychologist Rosenblatt (1958). The model was inspired by how the brain works. Containing a single neuron, the perceptron algorithm is the simplest neuronal network classifier. Simply put, a perceptron model is a neuronal network in which a single neuron receives inputs, that is, stimuli from other neurons through connections. Based on the magnitude of the activation of the stimuli, the neuron sends a signal.

In an NLP classification task, the input data x represents a set of stimuli, that is, features, $f_{1:d}$. In a binary classification, a feature f_d is mapped to a binary activation status $\{0, 1\}$ and has the weight, w_d. The weight represents the strength of the connection to a neuron. By combining features and weights linearly, we measure the magnitude of the activation of input stimuli on the neuron:

$$A = \sum_{d=1}^{D} w_d f_d = w_1 f_1 + w_2 f_2 + \cdots + w_d f_d \tag{G.46}$$

Based on the value of activation A, the neuron then sends a positive or negative signal to other neurons. In a binary classification task, the value of the signal corresponds to the output $y \in \{+1, -1\}$, that is, a positive class or a negative class. A binary classifier follows a decision rule (cf. Equation (G.47)). The signal indicates a positive class if the value of activation (A) is positive. The signal assigns a negative class if the value of activation (A) is negative.

$$signal(A) = y = \begin{cases} +1 & \text{if A} > 0 \\ -1 & \text{if A} < 0 \end{cases} \tag{G.47}$$

[7] In addition, I used the course materials by Collins `http://www.cs.columbia.edu/~mcollins/` (last visited: 2015/06/30)

Geometrically, each input instance x, which is represented by a d-dimensional feature vector, corresponds to a data point in a d-dimensional space. Each dimension corresponds to a feature. In a two-class classification, the space is divided into two spaces — a positive class and a negative class — by a line known as the hyperplane. The hyperplane is the line in a geometric space where the activation score A is zero (cf. Equation (G.48)). Because the hyperplane belongs to neither the positive class nor the negative class, and because it is the decision boundary between a positive or a negative class, the goal of perceptron learning is to find the optimal hyperplane that separates the inputs into a positive class zone or a negative class zone. Geometrically, the weight vector $w_{1:d} = [w_1, w_2 \ldots, w_d]$ is orthogonal to the hyperplane. Therefore, the goal of perceptron learning is de facto to find the weight vector of the linear function of Equation (G.48).

$$A = w_1 f_1 + w_2 f_2 + \cdots + w_d f_d = 0 \tag{G.48}$$

In addition, the activation value A is often formulated with the bias term b (cf. Equation (G.49)). By introducing the bias term b, the decision rule (G.47) will be reformulated with the threshold b (cf. Equation (G.50)). Without the bias term b, the hyperplane goes through the origin. By introducing a bias, the intercept is changed.

$$A = \sum_{d=1}^{D} w_d f_d + b \tag{G.49}$$

$$signal(A) = y = \begin{cases} +1 & \text{if } \sum_{d=1}^{D} w_d f_d > \text{b} \\ -1 & \text{if } \sum_{d=1}^{D} w_d f_d < \text{b} \end{cases} \tag{G.50}$$

The structured perceptron (Collins, 2002) in which we are interested is an extension of the standard two-class perceptron algorithm. The structured perceptron consists of two main components: 1) the local feature function; and 2) the global feature function. The local feature function is an indicator function that maps the features (i.e. the observation-target pairs) of input token x_n to a d-dimensional feature vector containing the value {0, 1} as MEMM, CRF, and the standard perceptron. Each input sequence $x_{1:N}$ has a feature representation by summing each local feature vector of N tokens in the sequence (cf. Equation (G.51)). The feature $g_d(x_{1:N}, y'_{1:N})$ counts the occurrence of features in the sequence. Each feature scalar is weighted and linearly combined to obtain a global score in (G.52). This can be decomposed as (G.53).

$$g_d(x_{1:N}, y'_{1:N}) = \sum_{n=1}^{N} f(x_n, y_n) \tag{G.51}$$

$$\Phi(x_{1:N}, y'_{1:N}) = \sum_{d=1}^{D} w_d g_d(x_{1:N}, y'_{1:N}) \tag{G.52}$$

$$\Phi(x_{1:N}, y'_{1:N}) = \sum_{n=1}^{N} \sum_{d=1}^{D} w_d f_d(x_n, y_n) \tag{G.53}$$

Decoding Because each input token x_n has K possible output classes $y_n^{1:K}$ to be mapped, we need to find the highest global score of the output sequence $y'_{1:N}$:

$$\underset{y'_{1:N} \in GEN(x_{1:N})}{\arg\max} \Phi(x_{1:N}, y'_{1:N}) \tag{G.54}$$

A Viterbi-like algorithm is used in Collins (2002) and beam search in Zhang and Clark (2011). In Viterbi-like decoding, we define the path score of each token (cf. Equation (G.55)). The path score is the maximal value of the multiplication of (i) the path score of the previous output y_{n-1} by (ii) the path score of the current token. The best scoring previous output y_{n-1}^{j} among J possible outputs at time t_{n-1} is backtracked by the back-pointer (BP) (cf. Equation (G.56)).

$$\pi(n, y_{n-1}^{j}, y_n^{k}) = \max_{0 \le j \le J} \pi(n-1, y_{n-2}^{i}, y_{n-1}^{j}) + \phi(x_n, y_n^{k}) \tag{G.55}$$

$$BP(n, y_{n-1}^{j}, y_n^{k}) = \underset{1 \le j \le J}{\arg\max}\, \pi(n-1, y_{n-2}^{i}, y_{n-1}^{j}) + \phi(x_n, y_n^{k}) \tag{G.56}$$

Parameter Estimation and Learning The perceptron algorithm is error-driven. The parameters are not changed unless the current model predicts a wrong class. If the current model predicts a wrong class, then the parameters are updated. Otherwise, they remain unchanged. In online perceptron learning, each training example x_d is classified by the current model. If the predicted class z_d is wrong, that is, not equal to the gold class y_d, the current weight vector $\mathbf{w}_d$ is updated as (G.57). Because a feature is an indicator function and its value is {0, 1}, the updating is carried out simply by adding 1 to the parameters of the features that lead to the gold standard tag y_d and by subtracting 1 from the parameters of features that lead to the wrong prediction z_d. In the former case, the parameters were too low, whereas they were too high in the latter case. The current weight vector $\mathbf{w}_d$ changes the direction to the correct input feature vector $\mathbf{f}_d(x_d, y_d)$ by addition and moves away from the incorrect input feature vector $\mathbf{f}_d(x_d, z_d)$ by subtraction.

$$\mathbf{w}_{d+1} = \mathbf{w}_d + \mathbf{f}_d(x_d, y_d) - \mathbf{f}_d(x_d, z_d) \tag{G.57}$$

In the average structured perceptron (Collins, 2002, Freund and Schapire, 1999), each weight vector $\mathbf{w}_s^t$ is summed and averaged. In other words, the weight vector is averaged over the number of sequences, S, of all iterations of time T as (G.58). Hence, the parameters of the model are normalized regarding the number of times the parameters retain the values of the current model during online learning (Daumé III, 2014).

$$\mathbf{w}_{avr} = \frac{1}{ST} \sum_{s=1,t=1}^{S,T} \mathbf{w_s^t} \tag{G.58}$$

In summary, this appendix discussed state-of-the-art statistical tagging methods to understand how input information is integrated into models and how optimal output sequences are found in the models.

Bibliography

Abney, S. (1996). Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344.

Abney, S. (1997). Part-of-speech tagging and partial parsing. In Young, S. and Bloothooft, G., editors, *Corpus-Based Methods in Language and Speech Processing*, volume 2, pages 118–136. Springer Netherlands.

Adafre, S. F. (2005). Part of speech tagging for Amharic using conditional random fields. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 47–54.

Aissen, J. (2002). Differential object marking: Iconicity vs. economy. *Natural Language and Linguistic Theory*, 21(3):435–483.

Arnold, J. E., Wasow, T., Losongco, A., and Ginstrom, R. (2000). Heaviness vs. newness: The effects of structural complexity and discourse status on constituent ordering. *Language*, 76(1):28–55.

Attardi, G., Simi, M., and Zanelli, A. (2013). Domain adaptation by active learning. In Magnini, B., Cutugno, F., Falcone, M., and Pianta, E., editors, *Evaluation of Natural Language and Speech Tools for Italian*, pages 77–85. Springer, Berlin, Heidelberg.

Aubin, S., Nazarenko, A., and Nedellec, C. (2005). Adapting a general parser to a sublanguage. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 89–93.

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–362.

Baayen, H. (2011). Corpus linguistics and naive discriminative learning. *Brazilian Journal of Applied Linguistics*, 11(2):295–328.

Bangalore, S. and Joshi, A. K. (1999). Supertagging: an approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Bayerl, P. (2004). Disambiguierung deutschsprachiger Diskursmarker: Eine Pilot-Studie. *Linguistik Online*, 18.

Becker, M. and Frank, A. (2002). A stochastic topological parser for German. *Proceedings of the 19th International Conference on Computational linguistics*, pages 71–77.

Behaghel, O. (1932). *Deutsche Syntax: Eine Geschichtliche Darstellung.* Carl Winters Universitätsbuchhandlung, Heidelberg.

Bever, T. G. (1970). The cognitive basis for linguistic structures. In Hayes, J. R., editor, *Cognition and the Development of Language*, pages 279—362. Wiley, New York.

Biber, D. (1988). *Variation across Speech and Writing.* Cambridge University Press, New York.

Blunsom, P. and Baldwin, T. (2006). Multilingual deep lexical acquisition for HPSGs via supertagging. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 164–171.

Bögel, T., Hautli-Janisz, A., Sulger, S., and Butt, M. (2014). Automatic detection of causal relations in German multilogs. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language*, pages 20–27.

Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING 2010*, pages 89—97.

Boland, J. E. (1997). Resolving syntactic category ambiguities in discourse context: Probabilistic and discourse constraints. *Journal of Memory and Language*, 36(4):588–615.

Brants, T. (2000). TnT – A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP)*, pages 224–231.

Braun, C. (2003). Parsing German text for syntacto-semantic structures. In *Prospects and Advances in the Syntax/Semantics Interface, Lorraine- Saarland Workshop Series*, pages 99–102.

Bredenkamp, A., Crysmann, B., and Petrea, M. (2000). Looking for errors: A declarative formalism for resource-adaptive language checking. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC).*

Bresnan, J., Cueni, A., Nikitina, T., and Baayen, H. (2007). Predicting the dative alternation. In Bouma, G., Kramer, I., and Zwarts, J., editors, *Cognitive Foundations of Interpretation*, pages 69–94. Royal Netherlands Academy of Science, Amsterdam.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing (ANLC)*, pages 152–155.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Buchholz, S. (1998). Distinguishing complements from adjuncts using memory-based learning. In *Proceedings of the ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing*.

Buchholz, S. and Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*.

Bundesamt für Justiz, editor (2007). *Gesetztesgebungsleitfaden: Leitfaden für die Ausarbeitung von Erlassen des Bundes*. Bern, 3 edition.

Candito, M., Anguiano, E. H., and Seddah, D. (2011). A word clustering approach to domain adaptation: Effective parsing of biomedical texts. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 37–42.

Carreras, X., Collins, M., and Koo, T. (2008). TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL)*, pages 9–16.

Carreras, X., Màrquez, L., and Padró, L. (2003). Learning a perceptron-based named entity chunker via online recognition feedback. In *Proceedings of the seventh conference on Natural language learning (HLT-NAACL)*, volume 4, pages 156–159.

Carreras, X., Surdeanu, M., and Màrquez, L. (2006). Projective dependency parsing with perceptron. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, pages 181–185.

Chen, J. (2010). Semantic labeling and parsing via tree-adjoining grammars. In Bangalore, S. and Joshi, A. K., editors, *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*, pages 431–447. The MIT Press.

Chen, M. and Zechner, K. (2011). Computing and evaluating syntactic complexity features for automated scoring of spontaneous non-native speech. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, pages 722—-731.

Chen, S. F. and Goodman, J. (1998). An Empirical Study of Smoothing Techniques for Language Modeling. Technical report, Harvard University.

Cheung, J. C. K. and Penn, G. (2009). Topological field parsing of German. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 64–72.

Choi, H.-W. (1996). *Optimizing the Structure in Context: Scrambling and Information Structure*. PhD thesis, Stanford University.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge (Massachusetts).

Christodoulopoulos, C., Goldwater, S., and Steedman, M. (2010). Two decades of unsupervised POS induction: how far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 575–584.

Ciaramita, M. and Altun, Y. (2006). Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 594–602.

Clark, S. (2011). Supertagging for combinatory categorial grammar. In *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, pages 19–24.

Clark, S. and Curran, J. R. (2007). Perceptron training for a wide-coverage lexicalized-grammar parser. In *Proceedings of the Workshop on Deep Linguistic Processing (DeepLP)*, pages 9–16.

Cohen, S. B. and Smith, N. A. (2007). Joint morphological and syntactic disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 208—217.

Coleman, E. B. (1962). Improving comprehensibility by shortening sentences. *Journal of Applied Psychology*, 46(2):131–134.

Coleman, E. B. and Blumenfeld, J. P. (1963). Cloze scores of nominalizations and their grammatical transformations using active verbs. *Psychological Reports*, 13(3):651–654.

Collins, M. (2002). Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing (EMNLP)*, volume 10, pages 1–8.

Collins, M. and Roark, B. (2004). Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 111–118.

Crysmann, B., Bertomeu, N., Adolphs, P., Flickinger, D., and Klüwer, T. (2008). Hybrid processing for grammar and style checking. In *Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1, pages 153–160.

Crysmann, B., Frank, A., Kiefer, B., Müller, S., Neumann, G., Piskorski, J., Schäfer, U., Siegel, M., Uszkoreit, H., Xu, F., Becker, M., and Krieger, H.-U. (2002). An integrated architecture for shallow and deep processing. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 441–448.

Curran, J. R., Clark, S., and Vadas, D. (2006). Multi-tagging for lexicalized-grammar parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 697–704.

Darroch, J. N. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480.

Daum, M., Foth, K. A., and Menzel, W. (2003). Constraint based integration of deep and shallow parsing techniques. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics (EACL)*.

Daumé III, H. (2014). *A course in machine learning*. http://ciml.info/dl/v0_9/ciml-v0_9-all.pdf.

Dell'Orletta, F., Montemagni, S., and Venturi, G. (2011). READ-IT: Assessing readability of Italian texts with a view to text simplification. In *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, pages 73–83.

Dipper, S. and Stede, M. (2006). Disambiguating potential connectives. In *Proceedings of KONVENS 2006 Workshop on the Lexicon-Discourse Interface*, pages 167–173.

Drach, E. (1963). *Grundgedanken der deutschen Satzlehre*. Wissenschaftliche Buchgesellschaft, Darmstadt.

Dubey, A. (2005). What to do when lexicalization fails: Parsing German with suffix analysis and smoothing. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 314–321.

Duchier, D. and Debusmann, R. (2001). Topological dependency trees: A constraint-based account of linear precedence. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 180–187.

Dudenredaktion (1999). *Duden Das grosse Wörterbuch der deutschen Sprache*. Dudenverlag, Mannheim, 3 edition.

Dürscheid, C. (2012). *Syntax. Grundlagen und Theorien*. Vandenhoeck & Ruprecht.

Eisenberg, P., Gelhaus, H., Wellmann, H., Henne, H., and Sitta, H. (1995). *Duden, Grammatik der deutschen Gegenwartssprache*. Dudenverlag, Mannheim, 5 edition.

Ekbal, A., Haque, R., and Bandyopadhyay, S. (2007). Bengali part of speech tagging using conditional random field. In *Proceedings of Seventh International Symposium on Natural Language Processing (SNLP)*, pages 131–136.

Eroms, H.-W. (2008). *Stil und Stilistik*. Erich Schmidt Verlag, Berlin.

Ezeiza, N., Alegria, I., Arriola, J. M., Urizar, R., and Aduriz, I. (1998). Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th International Conference on Computational Linguistics (COLING)*, pages 379–379.

Falkenjack, J. and Jönsson, A. (2014). Classifying easy-to-read texts without parsing. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 114—122.

Faruqui, M. and Padó, S. (2010). Training and evaluating a German named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*.

Feng, L., Elhadad, N., and Huenerfauth, M. (2009). Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 229—-237.

Feng, L., Jansche, M., and Huenerfauth, M. (2010). A comparison of features for automatic readability assessment. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 276–284, Beijing, China.

Ferraro, J. P., Daumé III, H., DuVall, S. L., Chapman, W. W., Harkema, H., and Haug, P. J. (2013). Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation. *JAMIA*, 20(5):931–939.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 363–370.

Finkel, J. R., Kleeman, A., and Manning, C. D. (2008). Efficient, feature-based, conditional random field parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 8, pages 959–967.

Flack, K. (2007). Ambiguity avoidance as contrast preservation: case and word order freezing in Japanese. *University of Massachusetts Occasional Papers in Linguistics: Papers in Optimality Theory III*, 32:57–88.

Fortmann, C. and Forst, M. (2004). An LFG grammar checker for CALL. In *Proceedings of InSTIL/ICALL2004 – NLP and Speech Technologies in Advanced Language Learning Systems*.

Foster, J. (2010). "cba to check the spelling" investigating parser performance on discussion forum posts. In *Proceedings of Human Language Technologies: 2010 Annual Conference of the North American Chapter of the ACL (HLT)*, pages 381–384.

Foster, J., Wagner, J., Seddah, D., and van Genabith, J. (2007). Adapting WSJ-trained parsers to the British National Corpus using in-domain self-training. In *Proceedings of the 10th Conference on Parsing Technologies*, pages 33–35.

Foth, K., By, T., and Menzel, W. (2010). Guiding a constraint dependency parser with supertags. In Bangalore, S. and Joshi, A. K., editors, *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*. MIT Press, Cambridge, Massachusetts and London, England.

Foth, K. A. (2005). *Eine umfassende Constraint-Dependenz-Grammatik des Deutschen*. University of Hamburg, Hamburg.

Foth, K. A. and Menzel, W. (2006). Hybrid parsing: using probabilistic models as predictors for a symbolic parser. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL)*, pages 321–328.

Frank, A., Becker, M., Crysmann, B., Kiefer, B., and Schäfer, U. (2003). Integrated shallow and deep parsing: TopP meets HPSG. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*.

Frase, L. (1983). Human factors and behavioral science: The UNIX Writer's Workbench software: Philosophy. *The Bell System Technical Journal*, 62(6):1883–1883.

Frazier, L. (1979). *On Comprehending Sentences: Syntactic Parsing Strategies*. PhD thesis, University of Connecticut.

Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. In *Image, Language, Brain*, pages 95–126. MIT Press, Cambridge, MA.

Giesbrecht, E. and Evert, S. (2009). Is part-of-speech tagging a solved task? An evaluation of POS taggers for the German web as corpus. In *Proceedings of the Fifth Web as Corpus Workshop (WAC5)*, pages 27–35.

Gildea, D. (2001). Corpus variation and parser performance. In *Proceedings of 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 167–202.

Goldberg, Y. and Tsarfaty, R. (2008). A single generative model for joint morphological segmentation and syntactic parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, pages 371–379.

Goutam, R. (2012). Exploring self-training and co-training for Hindi dependency parsing using partial parses. In *Proceedings of the 2012 International Conference on Asian Language Processing (IALP)*, pages 37–40.

Goutam, R. and Ambati, B. (2012). Exploring self-training and co-training for dependency parsing. In *Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 123—-134.

Green, N. D. and Žabokrtský, Z. (2012). Hybrid combination of constituency and dependency trees into an ensemble dependency parser. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data (HYBRID)*, pages 19–26.

Gries, S. T. and Wulff, S. (2013). The genitive alternation in Chinese and German ESL learners: towards a multifactorial notion of context in learner corpus research. *International Journal of Corpus Linguistics*, 18(3):327–356.

Haapalainen, M. and Majorin, A. (1994). GERTWOL: ein System zur automatischen Wortformerkennung deutscher Wörter. Technical report, Lingsoft, Inc.

Hagan, M. T., Demuth, H. B., Beale, M., and De Jesús, O. (2015). *Neural Network Design*. http://hagan.okstate.edu/NNDesign.pdf.

Hajič, J., Krbec, P., Květoň, P., Oliva, K., and Petkevič, V. (2001). Serial combination of rules and statistics: A case study in Czech tagging. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 268–275.

Hale, K. (1982). Preliminary remarks on configurationality. In *Proceedings of the 12th Annual Meeting of the North-Eastern Linguistics Society (NELS XII 1981)*, pages 86–96.

Hansen-Schirra, S. and Neumann, S. (2004). Linguistische Verständlichmachung in der juristischen Realität. In Lerch, K. D., editor, *Die Sprache des Rechts: Recht verstehen: Verstandlichkeit, Missverstandlichkeit und Unverstandlichkeit von Recht*, volume 1. Walter de Gruyter, Berlin.

Harbusch, K., Bäcker, J., and Hasan, S. (2010). Applications of HMM-based supertagging. In Bangalore, S. and Joshi, A. K., editors, *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*, pages 449–477. The MIT Press, Cambridge, Massachusetts London, England.

Harper, M. P. and Wang, W. (2010). Constraint dependency grammars: SuperARVs, language modeling, and parsing. In Bangalore, S. and Joshi, A. K., editors, *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*. MIT Press, Cambridge, Massachusetts and London, England.

Harris, Z. S. (1962). *String Analysis of Sentence Structure*. Mouton, Der Hag.

Hauk, W. and Lötscher, A. (1994). Verständlichkeit von Gesetzen als Problem der Gesetzgebung. *LeGes: Gesetzgebung & Evaluation*, 5(2):91–99.

Henderson, J. C. and Brill, E. (1999). Exploiting diversity in natural language processing: combining parsers. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 187–194.

Hinrichs, E. W. (2005). Finite-state parsing of German. In Arppe, A., Carlson, L., Lindén, K., Piitulainen, J., Suominen, M., Vainio, M., Westerlund, H., and Yli-Jyrä, A., editors, *Inquiries into Words, Constraints and Contexts. Festschrift for Kimmo Koskenniemi on his 60th Birthday*, pages 35–44. CSLI Studies in Computational Linguistics ONLINE, Califorinia.

Hinrichs, E. W., Kübler, S., Müller, F. H., and Ule, T. (2002). A hybrid architecture for robust parsing of German. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*.

Hobbs, J. R. and Bear, J. (1990). Two principles of parse preference. In *Proceedings of the 13th Conference on Computational Linguistics (COLING)*, pages 162–167.

Höfler, S. (2011). Ein Satz - eine Aussage. Multipropositionale Rechtssätze an der Sprache erkennen. *LeGes: Gesetzgebung & Evaluation*, 22(2):259–279.

Höfler, S. and Piotrowski, M. (2011). Building corpora for the philological study of Swiss legal texts. *Journal for Language Technology and Computational Linguistics (JLCL)*, 26(2):77–89.

Höfler, S. and Sugisaki, K. (2012). From drafting guideline to error detection: Automating style checking for legislative texts. In *Proceedings of the Second Workshop on Computational Linguistics and Writing (CLW): Linguistic and Cognitive Aspects of Document Creation and Document Engineering*, pages 9–18.

Höfler, S. and Sugisaki, K. (2014). Constructing and exploiting an automatically annotated resource of legislative texts. In *The 9th Edition of the Language Resources and Evaluation Conference (LREC)*.

Höhle, T. (1986). Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985s*, pages 329—340.

Jiampojamarn, S., Kondrak, G., and Cherry, C. (2006). Biomedical term recognition with the perceptron HMM algorithm. In *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology (HLT-NAACL)*, pages 114–115.

Joachims, T. (1998). Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, pages 137–142.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Education International, Upper Saddle River, New Jersey, 2nd edition.

Karlsson, F., Voutilainen, A., Heikkila, J., and Anttila, A. (1995). *Constraint Grammar - A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin.

Kate, R. J., Luo, X., Patwardhan, S., Franz, M., Florian, R., Mooney, R. J., Roukos, S., and Welty, C. (2010). Learning to predict readability using diverse linguistic features. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 546–554.

Keenan, E. L. and Comrie, B. (1977). Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8(1):63–99.

Kermes, H. and Evert, S. (2002). YAC-A recursive chunker for unrestricted German text. *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*.

Khan, M., Dickinson, M., and Kübler, S. (2013a). Does size matter? text and grammar revision for parsing social media data. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 1–10.

Khan, M., Dickinson, M., and Kübler, S. (2013b). Towards domain adaptation for parsing web data. In *Proceedings of the 9th Conference on Recent Advances in Natural Language Processing (RANLP-13)*, pages 357–364.

Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47.

Klatt, S. (2002). Combining a rule-based tagger with a statistical tagger for annotating German texts. In *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)*.

Klein, S. and Simmons, R. F. (1963). A computational approach to grammatical coding of English words. *Journal of the ACM*, 10(3):334–347.

Klenner, M. and Tuggener, D. (2011). An incremental entity-mention model for coreference resolution with restrictive antecedent accessibility. In *Recent Advances in Natural Language Processing (RANLP 2011)*, pages 178–185.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge (Massachusetts), London (England).

Koo, T., Carreras, X., and Collins, M. (2008). Simple semi-supervised dependency parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT)*, pages 595–603.

Kroeger, P. (2004). *Analyzing Syntax: A Lexical-Functional Approach*. Cambridge University Press, Cambridge, England.

Kübler, S., Beck, K., Hinrichs, E., and Telljohann, H. (2010). Chunking German: An unsolved problem. *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 147–151.

Kübler, S., Hinrichs, E. W., and Maier, W. (2006). Is it really that difficult to parse German? In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 111–119.

Kübler, S., Ivanova, S., and Klett, E. (2008). Combining dependency parsing with PP attachment. In *Fourth Midwest Computational Linguistics Colloquium*.

Kübler, S. and Maier, W. (2013). Über den Einfluss von Part-of-Speech-Tags auf Parsing-Ergebnisse. *JLCL*, 28(1):17–44.

Kuno, S. and Kaburaki, E. (1977). Empathy and syntax. *Linguistic Inquiry*, 8(4):627–672.

Kunze, C. and Lemnitzer, L. (2007). *Computerlexikographie. Eine Einführung.* Güter Narr Verlag Tübingen, Tübingen.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML)*, pages 282–289.

Lee, H. (2001). *Optimization in Argument Expression and Interpretation: A Unified Approach.* PhD thesis, Stanford University.

Lee, J., Naradowsky, J., and Smith, D. A. (2011). A discriminative model for joint morphological disambiguation and dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, volume 1, pages 885–894.

Lenerz, J. (1977). *Zur Abfolge nominaler Satzglieder im Deutschen.* TBL Verlang Gunter Narr, Tübingen.

Li, Y., Duan, H., and Zhai, C. (2012). A generalized hidden Markov model with discriminative training for query spelling correction. In *Proceedings of the 35th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 611–620.

Li, Z. and Zhou, G. (2012). Unified dependency parsing of Chinese morphological and syntactic structures. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1445–1454.

Liu, X., Zhang, S., Wei, F., and Zhou, M. (2011). Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, volume 1, pages 359–367.

Lluís, X. and Màrquez, L. (2008). A joint model for parsing syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL)*, pages 188–192.

Loftsson, H. and Östling, R. (2013). Tagging a morphologically complex language using an averaged perceptron tagger: The case of Icelandic. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 105–119.

Longadge, R. and Dongre, S. (2013). Class imbalance problem in data mining review. *International Journal of Computer Science and Network (IJCSN)*, 2(1).

Lötscher, A. (1995). Der Stellenwert des Verständlichkeit in einer Hierarchie der kommunikativen Werte von Gesetzen. *Bulletin Suisse de Linguistique Appliquée*, 2:109–127.

MacDonald, N. H. (1983). The UNIX Writer's Workbench software: Rationale and design. *The Bell System Technical Journal*, 62(6):1891–1908.

Mahlow, C. and Piotrowski, M. (2010). Noun phrase chunking and categorization for authoring aids. In *Proceedings of KONVENS 2010*, pages 57–65.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL)*, pages 49–55.

Manning, C. D. (2011). Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, volume 1, pages 171–189.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts London, England.

Martschat, S. and Strube, M. (2014). Recall error analysis for coreference resolution. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2070—-2081.

McCallum, A., Freitag, D., and Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 591–598.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL) at HLT-NAACL 2003*, volume 4, pages 188–191.

McCallum, A. and Nigam, K. (2001). A comparison of event models for Naive Bayes text classification. In *AAAI/ICML-98 Workshop on Learning for Text Categorization*, pages 41–48.

McClosky, D. and Charniak, E. (2008). Self-training for biomedical parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (HLT-Short)*.

McClosky, D., Charniak, E., and Johnson, M. (2006). Effective self-training for parsing. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):173–187.

Miller, S., Guinness, J., and Zamanian, A. (2004). Name tagging with word clusters and discriminative training. In *Proceedings of HLT-NAACL*, volume 4, pages 337–342.

Moore, R. C. (2005). A discriminative framework for bilingual word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 81–88.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers (ACLShort)*, pages 220–224.

Moore, R. C., Yih, W.-t., and Bode, A. (2006). Improved discriminative bilingual word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL (ACL)*, pages 513–520.

Müller, F. H. (2005). *A Finite-State Approach to Shallow Parsing and Grammatical Functions Annotation of German*. PhD thesis, Universität Tübingen.

Müller, F. H. and Ule, T. (2002). Annotating topological fields and chunks and revising POS tags at the same time. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, volume 1, pages 1–7.

Müller, G. (1999). Optimality, markedness, and word order in German. *Linguistics*, 37(5):777—818.

Müller, G. (2002). Free word order, morphological case, and sympathy theory. In Fanselow, G. and Fery, C., editors, *Resolving Conflicts in Grammars*, pages 9–48. Buske (Sonderheft der Linguistischen Berichte), Hamburg.

Müller, G. (2006). *Elemente einer Rechtssetzungslehre*. Schulthess Verlag, 2 edition.

Nenkova, A., Chae, J., Louis, A., and Pitler, E. (2010). Structural features for predicting the linguistic quality of text: applications to machine translation, automatic summarization and human-authored text. In Krahmer, E. and Theune, M., editors, *Empirical Methods in Natural Language Generation: Data-oriented Methods and Empirical Evaluation*. Springer, Berlin, Heidelberg.

Neumann, G., Braun, C., and Piskorski, J. (2000). A divide-and-conquer strategy for shallow parsing of German free texts. In *Proceedings of the Sixth Conference on Applied Natural Language gurocessing (ANLC)*, pages 239–246.

Nguyen, L. M., Nguyen, H. T., Nguyen, P. T., Ho, T. B., and Shimazu, A. (2009). An empirical study of Vietnamese noun phrase chunking with discriminative sequence models. In *Proceedings of the 7th Workshop on Asian Language Resources (ALR)*, pages 9–16.

Nguyen, N. and Guo, Y. (2007). Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 681–688.

Nivre, J., Hall, J., Kübler, S., MacDonald, R., Nilsson, J., Riedel, S., and Yuret, D. (2007). The CoNLL 2007 shared task on dependency parsing. *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

Nivre, J., Hall, J., and Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC)*, pages 2216–2219.

Nivre, J. and McDonald, R. (2008). Integrating graph-based and transition-based dependency parsers. In *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT)*, pages 950–958.

Nussbaumer, M. (2002). Es gibt nichts Gutes, außer man tut es – Arbeit an der Verständlichkeit von Gesetzestexten in der Schweizerischen Bundeskanzlei. *Hermes - Journal of Linguistics*, 29:111–123.

Nussbaumer, M. (2009). *128. Rhetorisch-stilistische Eigenschaften der Sprache des Rechtswesens*, volume Part 2. Mouton de Gruyter.

Paroubek, P. (2007). Evaluating part-of-speech tagging and parsing. On the evaluation of automatic parsing of natural language. In Dybkjær, L., Hemsen, H., and Minker, W., editors, *Evaluation of Text and Speech Systems*. Springer Netherlands.

Patel, C. and Gali, K. (2008). Part-of-speech tagging for Gujarati using conditional random fields. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 117–122.

Pauls, A. and Klein, D. (2011). Faster and smaller n-gram language models. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, volume 1, pages 258–267.

Peng, F., Feng, F., and McCallum, A. (2004). Chinese segmentation and new word detection using conditional random fields. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING)*, pages 562–568.

Peng, F. and McCallum, A. (2004). Accurate information extraction from research papers using conditional random fields. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 329–336.

Pereira, F. C. N. (1985). A new characterization of attachment preferences. In Dowty, D. R., Karttunen, L., and Zwicky, A. M., editors, *Natural language parsing*, pages 307–319. Cambridge University Press.

Petersen, S. E. and Ostendorf, M. (2009). A machine learning approach to reading level assessment. *Computer Speech & Language*, 23(1):89–106.

Pitler, E. and Nenkova, A. (2008). Revisiting readability: A unified framework for predicting text quality. In *Proceedings of EMNLP*.

Prince, A. and Smolensky, P. (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell, Malden.

Pyysalo, S., Salakoski, T., Aubin, S., and Nazarenko, A. (2006). Lexical adaptation of link grammar to the biomedical sublanguage: A comparative evaluation of three approaches. In *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine (SMBM)*, pages 60–67.

Quirk, C. and Corston-Oliver, S. (2006). The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 62–69.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286.

Ravin, Y. (1988). Grammar errors and style weaknesses in a text-critiquing system. *IEEE Transactions on Professional Communication*, 31(3):108–115.

Regierungsrat des Kantons Bern, editor (2000). *Rechtsetzungsrichtlinien des Kantons Bern (RSR). Modul 4: Sprache*. Bern.

Regierungsrat des Kantons Zürich, editor (2005). *Richtlinien der Rechtssetzung*.

Reichart, R. and Rappoport, A. (2007). Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616—623.

Reuther, U. (1998). Controlling language in an industrial application. In *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW)*.

Roark, B., Saraclar, M., Collins, M., and Johnson, M. (2004). Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 47–54.

Rooth, M. (1992). A theory of focus interpretation. *Natural Language Semantics*, 1:75–116.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, (65):386—407.

Sak, H., Güngör, T., and Saraclar, M. (2007). Morphological disambiguation of Turkish text with perceptron algorithm. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, pages 107–118.

Sang, E. F. T. K. and Buchholz, S. (2000). Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning (CoNLL)*, volume 7, pages 127–132.

Sarkar, A. (2001). Applying co-training methods to statistical parsing. In *Proceedings of the 2nd Meeting of the North American Association for Computational Linguistics (NAACL)*, pages 175–182.

Schiehlen, M. (2002). Experiments in German noun chunking. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*.

Schiehlen, M. (2004). Annotation strategies for probabilistic parsing in German. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 390–396.

Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). Technical report, Universität Stuttgart, Universität Tübingen, Stuttgart, Germany.

Schmid, H. (1999). Improvements in part-of-speech tagging with an application to German. In Armstrong, S., Church, K., Isabelle, P., Manzi, S., Tzoukermann, E., and Yarowsky, D., editors, *Natural Language Processing Using Very Large Corpora*, volume 11 of *Text, Speech and Language Processing*, pages 13–26. Kluwer Academic Publishers, Dordrecht.

Schmidt, H. and Schulte im Walde, S. (2000). Robust German noun chunking with a probabilistic content-free grammar. In *Proceedings of the 18th Conference on Computational Linguistics (COLING)*, volume 2, pages 726–732.

Schmidt-Wigger, A. (1998). Grammar and style checking for German. In *Proceedings of the Second International Workshop on Control Language Applications (CLAW)*, pages 76–86.

Schneider, A. and Stede, M. (2012). Proceedings of KONVENS 2012. In *Ambiguity in German Connectives: A Corpus Study.*

Schwarm, S. E. and Ostendorf, M. (2005). Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL).*

Seeker, W. and Kuhn, J. (2011). On the role of explicit morphological feature representation in syntactic dependency parsing for German. In *Proceedings of the 12th International Conference on Parsing Technologies (IWPT)*, pages 58–62.

Seeker, W. and Kuhn, J. (2013). Morphological and syntactic case in statistical dependency parsing. *Journal Computational Linguistics*, 39(1):23–55.

Sekine, S. (1997). The domain dependence of parsing. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, pages 96–102.

Sennrich, R., Schneider, G., Volk, M., and Warin, M. (2009). A new hybrid dependency parser for German. In *Proceeding of the German Society for Computational Linguistics and Language Technology*, pages 115–124.

Sennrich, R., Volk, M., and Schneider, G. (2013). Exploiting synergies between open resources for German dependency parsing, POS-tagging, and morphological analysis. In *Recent Advances in Natural Language Processing (RANLP)*, pages 601–609.

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, pages 104–107.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL)*, pages 134–141.

Shieber, S. M. (1983). Sentence disambiguation by a shift-reduce parsing technique. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics (ACL)*, pages 113–118.

Siewierska, A. (1993). On the interplay of factors in the determination of word order. In Jacobs, J., Stechow, A. v., Sternefeld, W., and Vennemann, T., editors, *Syntax: An International Handbook of Contemporary Research*, pages 826–846. Walter de Gruyter, Berlin.

Silfverberg, M., Ruokolainen, T., Linden, K., and Kurimo, M. (2014). Part-of-speech tagging using conditional random fields: Exploiting sub-label dependencies for improved accuracy. In *Proceedings of Fifty-Second Annual Meeting of the Association for Computational Linguistics (ACL)*.

Skut, W. and Brants, T. (1998a). Chunk tagger - statistical recognition of noun phrases. In *ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing*.

Skut, W. and Brants, T. (1998b). A maximum-entropy partial parser for unrestricted text. In *Proceeding of the 6th Workshop on Very Large Corpora*.

Smith, N. A. (2011). *Linguistic Structure Prediction*. Synthesis Lectures On Human Language Technologies. Morgan & Claypool Publishers.

Soffritti, M. (2009). *Eindeutig uneindeutig: Fachsprachen – ihre Didaktik, ihre Übersetzung*, chapter Bedingung und Zeitangabe – wenn und verwandte Ausdrucksformen in deutschsprachigen Gesetzbüchern für Südtirol. Peter Lang, Frankfurt A. M.

Soffritti, M. (2013). *Diatopische Variation in der deutschen Rechtssprache*, chapter Diatopische Unterschiede im Ausdruck von Bedingungen in deutschsprachigen Gesetzbüchern. Frank & Timme, Berlin.

Spoustová, j. D., Hajič, J., Raab, J., and Spousta, M. (2009). Semi-supervised training for the averaged perceptron POS tagger. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL)*, pages 763–771.

Steedman, M., Osborne, M., Sarkar, A., Clark, S., Hwa, R., Hockenmaier, J., Ruhlen, P., Baker, S., and Crim, J. (2003). Bootstrapping statistical parsers from small datasets. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 331–338.

Steinbach, M. and Vogel, R. (1998). The dative – an oblique case. In *Linguistische Berichte*, volume 173, pages 65–90.

Sternefeld, W. (2006). *Syntax. Eine morphologisch motivierte generative Beschreibung des Deutschen*. Stauffenburg-Verlag, Tübingen.

Sugisaki, K. and Höfler, S. (2013a). Incremental morphosyntactic disambiguation of nouns in German-language law texts. In *ESSLLI-13 Workshop on Extrinsic Parse Improvement (EPI)*, Düsseldorf.

Sugisaki, K. and Höfler, S. (2013b). Verbal morphosyntactic disambiguation through topological field recognition in German-language law texts. In Mahlow, C. and

Piotrowski, M., editors, *Systems and frameworks for computational morphology*. Springer, Berlin, Heidelberg.

Surdeanu, M. and Manning, C. D. (2010). Ensemble models for dependency parsing: Cheap and good? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT)*, pages 649–652.

Sutton, C. and McCallum, A. (2011). An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

Tapanainen, P. and Voutilainen, A. (1994). Tagging accurately: Don't guess if you know. In *Proceedings of the Fourth Conference on Applied Natural Language Processing (ANLC)*, pages 47–52.

Telljohann, H., Hinrichs, E. W., Sandra, K., Heike, Z., and Kathrin, B. (2012). Stylebook for the Tübingen treebank of written German (TüBa-D/Z). Technical report, Universität Tübingen.

Tesnière, L. (1959). *Eléments de syntaxe structurale*. Klincksieck, Paris.

Thurmair, G. (1990). Parsing for grammar and style checking. In *Proceedings of the 13th conference on Computational Linguistics*, volume 2, pages 365–370.

Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70.

Trushkina, J. (2004). *Morpho-syntactic annotation and dependency parsing of German*. PhD thesis, Universität Tübingen.

Tsarfaty, R. (2006). Integrated morphological and syntactic disambiguation for modern Hebrew. In *Proceedings of the 21st International Conference on computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING ACL): Student Research Workshop*, pages 49–54.

Tsarfaty, R., Seddah, D., Goldberg, Y., Kübler, S., Candito, M., Foster, J., Versley, Y., Rehbein, I., and Tounsi, L. (2010). Statistical parsing of morphologically rich languages (SPMRL): What, how and whither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 1–12.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. In *Proceedings of the 10th Panhellenic conference on Advances in Informatics (PCI)*, pages 382–392.

Vajjala, S. and Meurers, D. (2014). Exploring measures of "readability" for spoken language: Analyzing linguistic features of subtitles to identify age-specific tv programs. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, volume 14.

van Halteren, H. (1999). Performance of taggers. In *Syntactic Wordclass Tagging*. Kluwer Academic Publishers, Dordrecht.

Veenstra, J., Müller, F. H., and Ule, T. (2002). Topological field chunking for German. In *Proceedings of the 6th Conference on Natural Language Learning (COLING)*, pages 1–7.

Venturi, G. (2008). Parsing legal texts. A contrastive study with a view to knowledge management applications. In *Workshop 'Semantic Processing of Legal Texts' (LREC)*.

Versley, Y. (2005). Parser evaluation across text types. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT)*.

Versley, Y. and Rehbein, I. (2009). Scalable discriminative parsing for German. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 134–137.

Vogel, R. (2004). Remarks on the architecture of optimality theoretic syntax grammars. In Blutner, R. and Zeevat, H., editors, *Optimality Theory and Pragmatics*, pages 211—28. Palgrave Macmillan, Hampshire.

Volk, M. and Schneider, G. (1998). Comparing a statistical and a rule-based tagger for German. In *Proceedings of KONVENS 1998*, pages 125–137.

vor der Brück, T., Hartrumpf, S., and Helbig, H. (2008). A readability checker with supervised learning using deep syntactic and semantic indicators. In *Proceedings of the 11th International Multiconference: Information Society (IS)*, pages 92–97.

Voutilainen, A. (1995a). How to recognize subjects in English. In Karlsson, F., Voutilainen, A., Heikkila, J., and Anttila, A., editors, *Constraint Grammar - A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin.

Voutilainen, A. (1995b). A syntax-based part-of-speech analyser. In *Proceedings of the Seventh Conference on European Chapter of the Association for Computational Linguistics (EACL)*, pages 157–164.

Voutilainen, A. (1996). Designing a parsing grammar. Technical report, University of Helsinki.

Voutilainen, A. and Tapanainen, P. (1993). Ambiguity resolution in a reductionistic parser. In *Proceedings of the Sixth Conference on European Chapter of the Association for Computational Linguistics (EACL)*, pages 394–403.

Wang, Y.-Y., Acero, A., Mahajan, M., and Lee, J. (2006). Combining statistical and knowledge-based spoken language understanding in conditional models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, pages 882–889.

Zeevat, H. (2006). Freezing and marking. *Linguistics*, 44(5):1095–1111.

Zhang, Y. and Clark, S. (2007). Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 840–847.

Zhang, Y. and Clark, S. (2011). Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151.

Zhang, Y. and Nivre, J. (2011). Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT): short papers*, pages 188–193.

Zhang, Y.-z., Matsuzaki, T., and Tsujii, J. (2009). HPSG supertagging: a sequence labeling view. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 210—213.

Zhang, Y.-Z., Matsuzaki, T., and Tsujii, J. (2012). Structure-guided supertagger learning. *Natural Language Engineering*, 18(02):205–234.

Zhao, H., Huang, C.-N., and Li, M. (2006). An improved Chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165.